DOCUMENT RESUME

ED 258 561 IR 011 716

TITLE International Conference Proceedings of the Association for the Development of Computer-Based Instructional Systems (26th, Philadelphia, Pennsylvania, March 25-28, 1985).

INSTITUTION Association for the Development of Computer-based Instructional Systems.

PUB DATE Mar 85

NOTE 283p.

AVAILABLE FROM Association for the Development of Computer-based Instructional Systems, Miller Hall 409, Western Washington University, Bellingham, WA 98225 ($30.00).

PUB TYPE Collected Works - Conference Proceedings (021) -- Viewpoints (120) -- Reports - General (140)

EDRS PRICE MF01/PC12 Plus Postage.

DESCRIPTORS *Computer Assisted Instruction; Computer Graphics; Computers; Conference Proceedings; Courseware; Disabilities; *Educational Research; *Educational Theories; Health Education; Home Economics; Individualized Instruction; *Instructional Innovation; Microcomputers; Programing Languages; Videodisc Recordings; Videotape Recordings

IDENTIFIERS *Interactive Video

ABSTRACT

The 53 papers in these proceedings focus on the use of computer-based learning. Individual papers are organized by special interest group and presented under 12 headings: (1) Computer-Based Training (11 entries); (2) Elementary, Secondary, Junior College (1 entry); (3) Educators of the Handicapped (3 entries); (4) Health Education (6 entries); (5) Home Economics Consortium (3 entries); (6) Implementation (2 entries); (7) Interactive Instructional Users Group (3 entries); (8) Interactive Video/Audio (9 entries); (9) Mini-Micro (5 entries); (10) Music Consortium (1 entry); (11) PLATO Users Group (1 entry); and (12) Theory and Research (8 entries). Papers range from three to five pages and each paper includes an abstract, description of a project and/or presentation of data, diagrams or illustrations, and a conclusion. Most papers also provide bibliographic references. (JB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

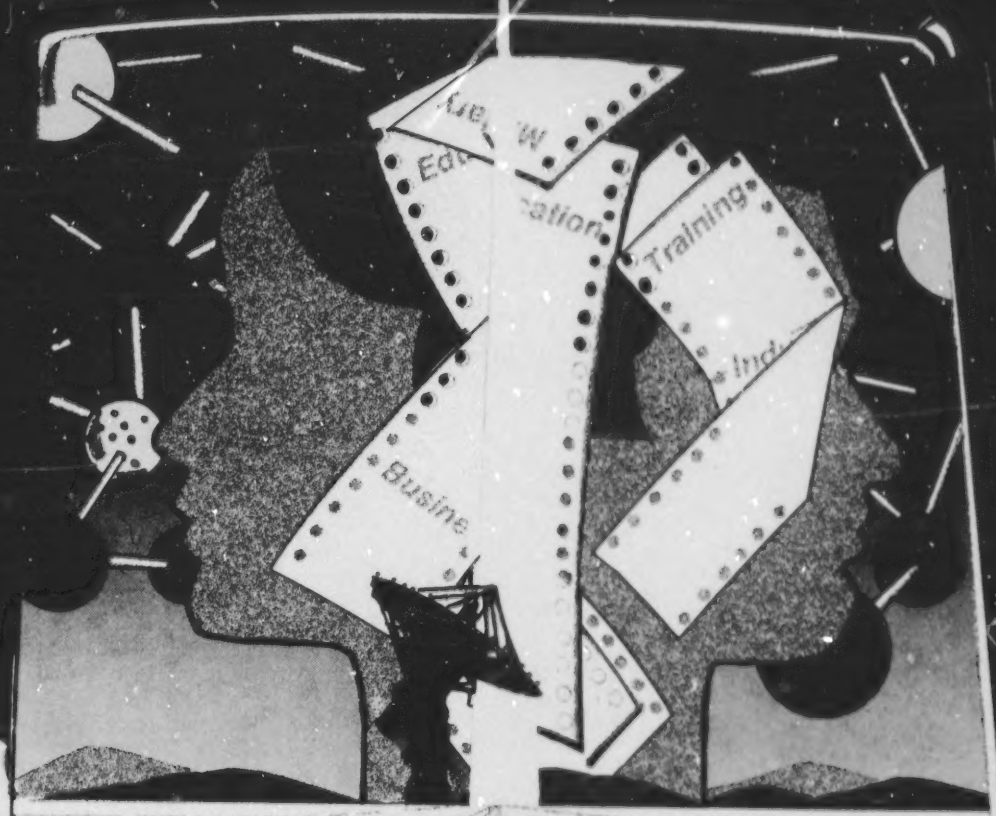
☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

26th International Conference Proceedings

March 25-28, 1985
Adams Mark Hotel
Philadelphia, Pennsylvania

ED258561



BEST COPY AVAILABLE

ADCIS

Association for the Development of Computer-Based Instructional Systems

IR011714

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

J.J. Hirschbuhl

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

26th ADCIS CONFERENCE PROCEEDINGS

Association for the Development of Computer-Based Instructional Systems

Philadelphia, Pennsylvania
March 25 thru 28, 1985

Copyright ADCIS 1985
ADCIS, MILLER HALL 409
WESTERN WASHINGTON UNIVERSITY
BELLINGHAM, WASHINGTON 98225

FORWARD

The use of computer-based learning is expanding at an ever-increasing rate. Schools, corporations and government agencies are finding more applications than the early CAI pioneers ever dreamed. ADCIS is the oldest organization dedicated to advancing the state-of-the art of computers and learning. This conference and these published proceedings contribute to the permanent record of the profession.

The work ADCIS members are doing to effect educational change across the world is of great importance. We are leading the charge of R. Buckminster Fuller when he called for "doing more with less." Our colleagues are dedicated to improving the effectiveness and efficiency of human learning across all ages and all peoples. The technological tools we are developing and testing offer great promise toward improving the prosperity of the world.

As you read these writings remember to thank those who take the time and have the energy to leave written records of our progress.

Glenn H. Head
Program Chairperson

ADCIS BOARD OF DIRECTORS

President
Ron Comer
The Ohio State University

Vice President-Treasurer
Jan Richardson
Gallaudet

Program Chairperson
Glenn Head
Instructional Communications

JCBI Editor
Allen Avner
University of Illinois

Newsletter Editor
Bill Heck
Goal Systems

Chapter Council Representative
Rick Anderson
University of Cincinnati

Special Interest Group Chairperson
Bob Tannenbaum
Hunter College

Special Interest Group Representative
Terris Wolff
University of Southern California

ADCIS Executive Director
Gordon Hayes
ADCIS International Headquarters
Ex-officio

SPECIAL INTEREST GROUP COUNCIL

Computer-Based Training
Robert Fratini
Western Electric

Elementary/Secondary/Jr. College/Math
Jeffry Gordon
University of Cincinnati

Educators for the Handicapped
Susan Rice
University of Minnesota

Health Education
Arthur D. King
University of Florida

Home Economics
Nancy Dillon
Phoenix, Arizona

Implementation
Terris Wolff
University of Southern California

Interactive Instructional System Presentation and Authoring
Sue Leslie
IBM Corporation

Interactive Video/Audio
Donna Larson
Grand Valley State Colleges

Mini/Micro
Bob Tannenbaum
Hunter College

Music
Michael Arenson
University of Delaware

Plato
Peter Johnston
University of Alberta

Theory & Research
Frank Roberts
Control Data Corporation

REGIONAL CHAPTERS

Front Range
Chuck Buchanan

Pacific Northwest
Ian Rose

Midwest
Chris Dimas

Mideastern
Rick Anderson

TABLE OF CONTENTS

COMPUTER-BASED TRAINING (SIG/CBT) **PAGE**

Using a Relational Data Base as a Means of Integrating Instructional and Library Materials in a Computer - Managed Course 1
John Franklin Arce, Alexander Joseph Romiszowski

How Computers Can Be Used to Support Training Decisions 5
Melissa Berkowitz, Ray S. Perez

A Comparative Study of Computer-Based Instruction vs. Lecture 10
John C. LeDoux, Charles J. Stanley

Project U-Train CAL - Entering the Home Stretch 15
Harry M. Nash

Effective Management of CAI Projects 18
Mary Jac Reed, George Reed

The Innovation Diffusion Process in Education - Will Computer-Based Instruction Survive? 24
Richard L. Reinhart

"GIGO"—How Does It Apply to Computer-Based Instruction? 29
Glenn E. Snelbecker, Debra S. Stepansky

Screen Design for CAI 34
Timothy W. Spannaus, Collette Pariseau

Computer-Based Instruction at the U.S. Army Engineer School Evolution and Management of an In-House Coding Team 38
Charles L. Stone

Screen Design Guidelines 42
William G. Sweeters

Orthographic Projection and Isometric Drawing Tutorial Using Computer Graphics 46
Manjula B Waldron, Mark Hopkins

ELEMENTARY, SECONDARY, JUNIOR COLLEGE (EL/SEC/JC/MATH)

Effectiveness of Phoenix as a Computer Based Education System 50
D.L. Jelden

EDUCATORS OF THE HANDICAPPED (SIG/HAN)

Computer Based Teacher Training 56
G. Phillip Cartwright, Cynthia N. Schloss

Model for Delivery of Braille - Instruction on a Statewide Basis 59
Marie Knowlton

An Interactive Microcomputer System for Teaching Reading and Writing to Young Communicatively Handicapped Children 63
Philip M. Prinz, Keith E. Nelson

HEALTH EDUCATION (HESIG)

A Microcomputer Tutorial EKG Principles 66
Kenneth J. Friedman, Joseph Boyle III, Joy Black, Gloria Robinson

A Profile of Learner Traits and Learning Outcomes with Computer-Assisted Instruction 71
Betty J. Paulanka

Computer Phobia: How to Help People Overcome It 76
Jane Timpke

A Survey Study of Computers in Nursing Education 79
Barbara Thomas

Promoting Computing in Nursing Education: a Delphi Study 85
Barbara Thomas

A Microcomputer Authoring System for Clinical Simulations 94
Richard S. Trynda

HOME ECONOMICS CONSORTIUM (SIG/HOMEC)

Microcomputers in Home Economics Curriculum in California Community Colleges 97
Marilyn D. Ambrose

Software: How Do You Chose? 102
Carol Byrd-Bredbenner, Suzanne Pelican

PAGE

State of the Art: Microcomputers in Home Economics Higher Education 104
Peggy S. Meszaros, M. Sue Stanley

IMPLEMENTATION (ISIG)

Microcomputers & Researchers: Information Retrieval 107
Frederick G. Knirk
Effective Manuals Are More Than Content 112
Patricia Anne Lawson

INTERACTIVE INSTRUCTIONAL SYSTEMS USERS GROUP (SIG/IIS)

IBM Pilot — Sophistication and Support 270
George Gerhold, Larry Kheriaty
CBT Development — Micro to Mainframe 116
Travis L. Piper
Designing Courseware Marcos 121
Timothy D. Taylor

INTERACTIVE VIDEO/AUDIO (SIG/IVA)

Implementation of a Software Interface With Bar Code Data Entry for an Intelligent Interactive Videodisc Instructional System 126
Gregory R. Bauer, Albert E. Miller, Eugene W. Henry
Report of a Pilot Study to Examine the Use of Interactive Videodisc Cardiopulmonary Resuscitation Instruction 135
Margaret J. A. Edwards, Kathryn J. Hannah
An Application of Microcomputer/Videodisc Instruction 141
Jerry W. Edwards, Gerald M. Gillmore, William K. Harmaning, Eric R. Halsey, Nana Lowell, Gary A. Kilgore, James M. Laffey
Interactive Video — The Next Frontier 144
Dorothy J. Fishman
From Novice to "Expert" In the Development of a Video Disc 149
Gloria Lombardi
Lesson Design, Authoring, and Video Issues in the Production of Interactive Videotapes 153
John F. Moore
Design of a Software Interface For an Intelligent, Interactive, and Integrated Videodisc System 157
Darleen Pigford, Albert E. Miller, Eugene W. Henry
An Interactive Video Curriculum Model 165
Mark Sanders
Computer Aided Instruction of Descriptive Geometry 169
Lester F. Smith Mark R. Hopkins, Jacob, J. Wolf III, Ding T. Jou

MINI-MICRO

Evaluation of Student Performance Using the Apple Microcomputer System 175
John H. Dustman
The Capabilities of the MicroTICCIT Authoring System 182
Jeffery J. Howell
Towards the Development of an IBM PC Authoring System 187
George Reed, Louisa Frank, Deborah Richards, Nancy Balogh
Analog to Digital Conversion 193
Robert S. Tannenbaum, David C. Tannenbaum
Configuring Your Personal Computer 203
Robert S. Tannenbaum

MUSIC CONSORTIUM (NCCBMI)

A Machine-Independent Data Structure for the Representation of Musical Concepts and Structures: Computer-Generated Musical Examples for CBI 208
J. Timothy Kolosick

PLATO USERS GROUP (PUG)

Providing Access to Computer-Based Education in Free Public Libraries 216
Clella B. Murray

THEORY AND RESEARCH (SIG/TAR) **PAGE**

The Computer as Transition From Object Manipulation to Abstract Visuo-Spatial Thought 222
Donna F. Berlin, Arthur L. White
Intelligent Tutoring of Physics Via Personal Computer — Emphasis on Learning 230
Stephen F. Brown, James Miller, Kathleen Swigger
Using Computer Graphics and Animation in Testing 238
James R. Okey, Michael E. Hale
On Cenceptualizing Research Methodologies For Computer-Based Instruction 241
David Reinking
Computer-Mediated Small-Group Discussion and its Potential for Data Collection 246
A.J. L. Sinclair
Research on User Interface Design as Related to Computer-Based Instruction 251
Mary L. Stoddard
Adaptive Testing by Computer: Benefits, Costs and Issues 256
J. Jon Veloski, Andrew R. Baggaley
Measuring the Effectiveness of a Computer-Based Self-Assessment Program 263
J. Jon Veloski, James Robinson, Mohammadreza Hojat

PROCEEDINGS INDEX

A	PAGE
AMBROSE, MARILYN	97
ARCE, JOHN FRANKLIN	1
B	
BAGGALEY, ANDREW	256 - 263
BALOGH, NANCY	187
BAUER, GREGORY R.	126
BERKOWITZ, MELISSA	5
BERLIN, DONNA	222
BLACK, JOY	66
BOYLE, JOSEPH III	66
BROWN, S.F.	230
BYRD-BREDBENNER, C.	102
C	
CARTWRIGHT, PHILLIP	56
D	
DUSTMAN, JOHN	175
E	
EDWARDS, JERRY	141
EDWARDS, MARGARET	135
F	
FISHMAN, DOROTHY J.	144
FRANK, LOUISA	187
FRIEDMAN, KENNETH J.	66
G	
GERHOLD, GEORGE	270
GILLMORE, GERALD	141
H	
HALE, MICHAEL E.	238
HALSEY, ERIC	141
HANNAH, KATHRYN	135
HARMANING, WM. K.	141
HENRY, EUGENE W.	126 - 157
HOJAT, MOHAMMADREZA	263
HOPKINS, MARK	46 - 169
HOWELL, JEFFERY J.	182
J	
JELDEN, D.L.	50
JOU, DING T.	169
K	
KHERIATY, LARRY	270
KILGORE, GARY A.	141
KOLOSICK, TIMOTHY	208
KNIRK, FREDERICK	107
KNOWLTON, MARIE	59
L	
LAWSON, PATRICIA ANNE	112
LAFFEY, JAMES M.	141
LE DOUX, JOHN C.	10
LOMBARDI, GLORIA	149
LOWELL, NANA	141
M	
MESZAROS, PEGGY	104

	PAGE
MILLER, ALBERT	126 - 157
MILLER, JAMES	230
MOORE, JOHN F.	153
MURRAY, CLELLA B.	216
N	
NASH, HARRY M.	15
NELSON, KEITH	63
O	
OKEY, JAMES	238
P	
PARISEAU, COLLETTE	34
PAULANKA, BETTY	71
PELICAN, SUZANNE	102
PEREZ, RAY	5
PIGFORD, DARLEEN	157
PIPER, TRAVIS	116
PIPNZ, PHILIP	63
R	
REED, GEORGE	18 - 187
REED, MARY JAC	18
REINHART, RIGHARD L.	24
REINKING, DAVID	241
RICHARDS, DEBORAH	187
ROBINSON, GLORIA	66
ROBINSON, JAMES	263
ROMISZOWSKI, JOSEPH	1
S	
SANDERS, MARK	165
SINCLAIR, A.J.L.	246
SCHLOSS, CYNTHIA	56
SMITH, LESTER	169
SNELBECKER, GLEN E.	29
SPANNAUS, TIMOTHY	34
STANLEY, CHARLES J.	10
STANLEY, SUE	104
STONE, CHARLES L.	38
STEPANSKY, DEBRA	29
STODDARD, MARY	251
SWEETERS, WILLIAM	42
SWIGGER, KATHLEEN	230
T	
TANNENBAUM, DAVID	193
TANNENBAUM, ROBERT	193 - 203
TAYLOR, TIMOTHY	121
THOMAS, BARBARA S.	79 - 85
TIMPKE, JANE	76
TRYNDA, RICHARD	94
V	
VELOSKI, JON	256
W	
WALDRON, MANJULA	46
WHITE, ART	222
WOLF, JACOB	169

USING A RELATIONAL DATA BASE AS A MEANS OF INTEGRATING INSTRUCTIONAL AND LIBRARY MATERIALS IN A COMPUTER - MANAGED COURSE

John Franklin Arce SENAI, Sao Paulo, Brazil

Alexander Joseph Romiszowski Concordia University

ABSTRACT

The paper describes a project installed in SENAI, a vocational training institution in Brazil. The principal characteristics of the project are the use of readily available hardware and software components as basis for a course management system that offers the capability of integrating the trainee's use of instructional materials with other generally available reference and study materials. The system promotes a student-directed approach to learning and offers great flexibility for use in teacher-led instruction or in fully or partly individualized systems. A further important aspect, discussed in the paper, is the methodology utilized to author the instructional materials, designed to permit the use of these materials either in printed paper format or by some means of electronic storage. The authoring methodology also is designed to facilitate easy and frequent updating and produces, as a by-product, a knowledge structure, or conceptual map, which permits easy access to the materials on a flexible, non-linear basis, controlled by the learner himself, instructors, and other advisors (e.g. librarians), or indeed by a computer-based tutorial system.

Although implemented at present in the subject area of electronics and in a relatively simple, part-paper, part-computer mode of operation, the basic principles may be applied in many diverse subject areas and are the basis for growth to a more sophisticated expert system implementation.

1. THE CONTEXT OF THE PROJECT

The project described in this paper is in operation at SENAI, in Sao Paulo, Brazil. SENAI is the National Service for Industrial and Apprenticeship Training, an organization which is financed and managed by Brazilian industry and is responsible for pre-service and in-service training of apprentices, adults, and skilled workers. Although SENAI is a national organization, it is decentralized in its mode of operation and each State of the Federation runs its own schools and offers courses that are adapted to the needs of local industry. The State of Sao Paulo is by far the most industrialized and is a leader in the development of new technologies, so it is natural for the SENAI of Sao Paulo to take the lead in the development of new training courses and systems that meet the needs of industries which are emerging or undergoing technological transformation.

Over the last few years, the Sao Paulo SENAI has been preparing itself to meet the growing demand for high level training in the fields of digital electronics, microprocessors, computer processing, robotics, and numerical control.

In December, 1983 a project entitled the "electronics project" was launched. This project is composed of sixteen sub-projects, which, apart from the development of training curricula and instructional materials, also include the development of simulators and trainers for the practical component of the training, a variety of computer-based aids and a programme of professional development for the SENAI staff engaged on the project. Three of the sub-projects are of especial relevance to this paper.

1. a series of five modular courses for the updating/retraining of electronics technicians in digital circuit design and maintenance in process control applications.
2. modular training in the maintenance of computer peripherals (involving precision mechanics and electronics)
3. modular training in electronics for maintenance electricians.

2. THE MODULAR TRAINING COURSES

The modular structure of these three training programmes was dictated by the characteristics of

the target population and their disparate needs, identified in the early stages of analysis. Each programme is composed of several modules of about two weeks in duration. Trainees may enter the programme at any point that is suited to their present needs and skills.

Each module has both a theoretical and a practical component, emphasis in the latter being placed on simulation of the tasks actually performed in the industrial context. The theoretical component is thoroughly documented in the form of structured, or "mapped" materials, suitable for both initial learning and later on-the-job reference. These materials are developed in accordance with the rules and suggestions of the INFORMATIONAL MAPPING (TM) system of structured writing (*), together with some additional rules developed especially for this project.

In addition to the use of these "maps," trainees are encouraged to make full use of other materials available in the training centre library, especially articles in specialist journals--often the most up-to-date sources of information in a fast-developing field like microelectronics.

3. THE INSTRUCTIONAL MATERIALS

The system of writing was designed to incorporate features that make it especially useful for the authorin of computer-based textual material. One of these features is the organization of the information in "maps." A map presents, in structured format, all the information deemed necessary on one specific topic. The title of the map clearly indicates the topic in question, thus facilitating the reader's access to relevant information. The information that is presented is closely analyzed and classified into types (e.g. definition, example, application, common synonyms, relation to other topics, etc.). Each different type of information is presented in a seperate and clearly marked-off "block" (paragraph) and is identified by a descriptive sub-title that, once more, facilitates the reader's selection of relevant information. The material may be read in a flexible, non-linear, manner, adapted to the user's needs and learning style.

Another characteristic of this system that is of especial value in the SENAI project, is the facility with which mapped materials may be updated or modified. The requirement for flexible, non-linear use implies that each block of information must be written in such a way that it makes sense on its own, when removed from the context of the material that immediately precedes and follows it. This feature, coupled with the division of the material into relatively short maps that deal with only one topic (or aspect of a topic), facilitate the processes of modification and updating of the content.

Yet another characteristic that we are exploiting in the SENAI project, is the use of clear map titles as indicators of the KEY WORDS that may be used to relate a given map to a given book chapter or article available in the library.

4. INTEGRATING THE LIBRARY INTO THE SYSTEM

As outlined above, trainees are expected to make intensive use of the library for a variety of reasons. They may use this resource to follow up topics of individual interest that go beyond the objectives of the module, or to gain further technical knowledge on topics dealt with in condensed form in the course, or to review concepts that were not fully understood in classroom sessions, or simply to access reference data not required to be learned but nevertheless necessary for the execution of practical assignments. The librarians have the task of indexing the existing materials in such a way that useful study prescriptions may be offerred to individual trainees.

In order to perform this task, the librarians require a structured set of key words that, as a whole, define the subject area in question (in our case, digital electronics and its applications). This key-word set is provided by the authors of the mapped materials, who take great pains to analyze the subject thoroughly and structure the maps so as to include all relevant key words in the map titles and sub-titles. This knowledge-structure serves as a guide to authors when writing the materials, as a guide to traineed when using the materials, and as a classification tool for librarians when indexing the other related materials available.

Once analyzed and indexed, the information is stored in a relational data base, together with data on the maps available at the present time (in the materials developed for all three of the sub-projects). The system thus enables the rapid localization of interrelated materials. A trainee interested in greater detail on a topic introduced in a given map, may receive a listing of all the library materials that deal with this topic. Another trainee, browsing through an article ofinterest, may encounter an unfamiliar concept or technical term and may call up a listing of the instructional materials available.

At present, the operation of the system is restricted to the librarians, who thus act as advisers to the trainees, using the data base as a primitive form of expert system. This, apart from reducing the hardware requirements (quantity and maintenance), ensures that the best possible service is offerred to the ultimate user (the trainee), as the librarian may quickly learn to apply diagnostic questioning procedures that identify with greater precision the trainee's exact information needs. However there is nothing that would preclude the direct use of the system by the trainees, if this proves to be desirable.

Thus, although not so used at present, the system has the basic characteristics necessary to operate as an individualized system of self-instruction, at least insofar as the theoretical component of the training is concerned. The practical component, being laboratory based, requires the presence of the instructor to perform tasks of orientation and evaluation.

5. COMPUTER MANAGED CONTROL AND EVALUATION

The evaluation of the practical activities is generally based on the instructor's observations of the trainees and the analysis of their final reports. These observations are, however, only one part of the total evaluation and control system. Entry into a particular laboratory activity is controlled by a pre-test which checks the trainee's mastery of pre-requisites that should have been learnt in preceeding modules or units. These tests are currently administered in normal, printed paper format. However the items are listed in a classified item bank in the data base, together with cumulative data on the use made of specific items, error frequencies and so on.

Thus, the data base enables the same test to act both as a pre-requisite test to control entry into laboratory activities and as a post-test that provides both summative and formative data to the instructor and other interested parties. For example, materials that require revision, instructors may use it as a powerful aid to diagnosis and treatment of specific individual learning difficulties and training administrators and client organizations may request a range of summative reports on the progress and overall outcomes of the training programmes.

6. THE SYSTEM SPECIFICATION

The system runs on a 6502 8-bit microprocessor machine, with a Z-80 enhancement running the CP/M operating system. The relational data-base management system used is D-BASE-2. The system includes the following:

--a series of data-base files for registering students, enquiries regarding the status of students, etc.

--a library data-base, organized on the basis of key-words thatare linked to the structure of the mapped instructional materials (as described earlier)

--a means of IDENTIFYING/CLASSIFYING test items and the use made of them (we do not store the full text of the test items in the system, due to limited storage capacity, difficulties associated with the graphics content of many of the items)

--a means of generating prescriptions based on the performance of individual students on a specific module or part of a module, indicating relevant maps/library materials.

With regard to the mass storage needs of the system, we are using a 10 megabyte hard disc. This, as indeed all the hardware components of our system, is Brazilian manufactured. The basis of the system is a commercially available microcomputer designed for small business and personal use, modified in some aspects to meet SENAI specifications dictated by the characteristics of this project. A critical aspect of maintaining a system of this nature is the method of back-up employed. In this respect, we are backing up the hard discs of our installed systems on a Burroughs 6900 system, of the National Commercial Apprenticeship and Training Service, with which SENAI has a specific arrangement.

Immediate plans include the linking of the eight currently available hard-disc systems in a local network and a remote network, in order to add the capabilities ofdialogue access by interested users and to use phone lines to download specific course systems to remotely located training schools, such as the SENAI school in Santos, currently used as the principal pilot site.

Since the programme is menu-driven, we have, in a typical interaction, an opening, or initial, menu which shows the different possible uses of the system - the library indexing file, data on students, queries, maps, questions, test items and evaluation results. Each one of these menu items has a password attached to it, restricting access to specific user groups. Each of these items, when selected, leads to a further, more specific list of menu options. For example, selection of the "student" option in the initial menu, would present us with the sub-file of options such as reports on the names of students registered, their companies of origin, the modules on which they are enrolled, the course group in which a given student is enrolled, a student's prior formal schooling and relevant experience, etc. Selection of the "evaluation" menu would lead to a sub-file containing options such as error rates on specific tests or items, performance data on specific students or groups, progress of students from a specific organization, and so on. It is possible, due to the hierarchical structure of the data base, to access the data stored in a large variety of ways, coming through the interrelationships specified in the various menus and the data files which are driven by these menus.

In summary, therefore, our system has many, if not all the characteristics of most systems for the computer-based management of instruction. One characteristic currently absent from our system, which many other systems (designed to run on much more powerful hardware) generally have, is the ability to store the test items and thus generate tests or offer on-line testing capabilities. This is outside the storage capabilities of our present hardware. Given the requisite expansion of storage, however, the D-BASE-2 software would be capable of handling this. On the other hand, the system does offer the rarely-met, perhaps unique, capability of integrating systematically developed instructional materials with other resources not so developed, knitting the combination into a system in which one set of materials complements and enriches the other. Due to this capability, the system immediately ceases to be totally bound by some set of pre-specified learning objectives, or by some pre-determined set of learning sequence and strategy alternatives.

7. POSSIBLE APPLICATIONS AND IMPLICATIONS

The system described and currently in operation can be characterized as economical, prac-

tically feasible and largely adequate for the requirements of the parent institution. It is seen to be economical in that it uses only readily available hardware and software, of specifications that may now be obtained "over the counter" as it were, in a growing number of countries that have even quite young microcomputer industries. It is seen to be practically feasible, from the experience of SENAI, Sao Paulo, in managing to take the described project from initial conception to full scale implementation in a matter of months (the project team was modest in number and largely new to projects of this nature). It is seen to be largely adequate by SENAI, designed as it was to meet the specific philosophy and set of constraints which govern the mode of operation of the institution.

However, the design methodology adopted is such as to permit a number of future developments, given appropriate hardware and software availability and, of course, institutional interest and committment. Many of the possible routes for development may not be of interest of special usefulness to SENAI, but may turn out to be of vital and great use in other institutional contexts. We therefore close this paper with just a brief list of some of the directions in which a project of this nature could be expanded.

Firstly, the instructional materials, as well as the test items, are so written that they may readily be stored and displayed means of some computer based system of delivery. The memory capacity required is, of course, immense and is not justified in many applications (printed text being preferable). However, in the case of wishing to provide access to the materials, on an individual basis and at a distance, the structured materials are already written in a style that permits the presentation and use of individual blocks of information as self-contained entities. There would be no great problems in adapting the material to a screen-by-screen presentation, largely independent of sequential interdependancies. This is an important characteristic of materials designed to be used as a dictionary or encyclopaedia --also of the information storage/presentation aspects of expert systems. It also permits the implementation of learner-directed learning strategies, inhibited by the more common programmed-instruction-based approaches to CAI.

Secondly, the special characteristic of this system, which enables the cross-referencing of instructional and other related library-based materials, may be of especial interest to teachers in many subject areas, not necessarily technical nor vocationally oriented. Indeed, this approach may, on face value, appear even more suited to subjects like history and philosophy, where there is a general difficulty in agreeing on specific, obligatory learning objectives, wh[illegible] immense source bibliographies (and often, [illegible]systematic aids to selection) exist and where the philosophy of learner autonumy has long been espoused (if little practiced due to practical constraints).

Thirdly, one must emphasize the importance of the "by-product" of a structual representation of the subject matter, in the form of a key-word hierarchy or network. Apart from its use in the current project as an authoring job-aid and a materials analysis/classification tool, it may serve as the basis for yet other system enhancements. One we have mentioned already is the use of the structure by the trainee as a structural guide and advance organizer of learning strategies. Another, of particular importance to CAI, is the value of the structure as a basis for the transformation of the present system into a true "expert system," capable not only of presenting information on request, but also of diagnosing the user's information needs and offering tutorial guidance.

(*) The names INFORMATIONAL MAPPING and INFOMAPS are registered trade names of Information Mapping Inc., Lexington, Massachussetts. The system was originally developed by Robert Horn and his associates (see Horn, 1974).

HOW COMPUTERS CAN BE USED TO SUPPORT TRAINING DECISIONS

Melissa S. Berkowitz and Ray S. Perez

US Army Research Institute

ABSTRACT

The purpose of this paper is to describe tools which provide Army training managers with information for making training decisions. Training decisions can range in scope from which soldier needs experience on a specific job task to determining training requirements for the readiness of all of the Army's soldiers. The evolution of computers and data base technology makes the collection, analysis, and presentation of vast quantities of information available to all levels of managers. This paper will present one approach for the development of Decision Support Systems (DSS) that are designed to support managers in the utilization of information. It will also discuss two prototype Army information systems.

DECISION SUPPORT SYSTEM CONCEPT

A Decision Support System (DSS) is composed of three subsystems: data, decision models, and the decision maker (Sprague and Watson, 1977b). Data are observations of the environment which are collected, coded, and stored. A building block of the DSS is a Management Information System which consists of a data base of observations, a procedure for the collection of data, a structure for the analysis and storage of data, an inquiry system for access to the data, and retrieval structure for reporting data analyses (Sprague and Watson, 1977a). Decision models are used to identify inferences and predictions of action alternatives from the data base analyses. Action alternatives are evaluated with respect to the goals and constraints of the decision requirements. Decision models may also indicate a criterion of choice and simulate the selection of an action and the impact of resources committed to implementing a decision (Kriebel, 1975). Advances in computer technology and management theory have made the data and decision models subsystems of DSS widely available. However, the utility of DSS relies largely with the third subsystem - the decision maker. Sprague and Watson (1977a) have observed limited utility of DSS because managers fail to use them. They indicate that decision models are typically not interrelated, rely on new data collection for the use of each model, and that the output from each model is not fed back to the data base or to other models. These problems must be remedied before decision makers will come to rely on DSS. Another factor that affects the use of DSS is that the analytic methods used in decision models are not transparent to the user. Also the system's interface between user and system may not well designed making the system difficult for novice computer users.

DESIGNING A DECISION SUPPORT SYSTEM

Rochart and Tracy (1980) suggest a four step approach to the design of a training DSS. The first step is to define the decision making functions of the training managers. The second step is to identify and review existent training data bases for relevance, validity, and meaningfulness. The third step is to select and/or design the decision making model that

survey indicating prior experience on specific maintenance tasks. Also all maintenance tasks performed within the unit are credited to the mechanic who performed them. Supervisors assign inexperienced mechanics to work with lead mechanics so that the inexperienced mechanics can get on-the-job experience. Supervisors may also set up special training sessions on selected tasks. The Skill Development component also provides a Workshop to prepare maintenance trainers.

Mechanic Certification. The purpose of Mechanic Certification is to recognize mechanics who can demonstrate technical proficiency in specified maintenance tasks. The record of the qualified mechanic is reviewed by the supervisor to ensure that the mechanic's performance meets the requirements of Certification. An award ceremony is scheduled and the Notice of Certification is entered in the mechanic's military record.

Training Information Management Model-Executive

TIMM-E is a DSS designed to help training managers make decisions effecting training policy and evaluation. The decision making functions of TIMM-E are primarily concerned with critical factors that determine the success or failure of training activities. Typical questions answered by TIMM-E are: How well does training occur? How are training resources utilized? and What is the relationship between training received and performance on the job?

TIMM-E is a menu-driven system developed on a mainframe computer. The user interacts with the system by using English language commands to query specific training data bases. TIMM-E guides the user through a sequence of problem solving steps. Each step considers policy and evaluation issues and relates these issues to appropriate training information. As the user progresses through each step he/she is provided with descriptive data. That enables him/her to assess the status of training programs and identify the factors that affect the status of a training activity. At the conclusion of these problem solving steps the user is presented a recommendation and rationale for the recommendation, along with an estimate of the impact of alternative action on the status of the training program and available resources. Recommendations are made on the basis of a best fit model, one that satisfies all the conditions and concerns of the user.

TIMM-E supports five modes of inquiries that are designed to help the user formulate a decision. These are: describe, compare, explain, predict, and decide. These inquiries also represent the sequence of the problem solving steps of the decision making model embedded in the system. The following sections will provide descriptions of each of the TIMM-E inquiry modes.

Describe inquiry mode provides the user with descriptive data that characterizes the status of people, places, things, or activities. The user is prompted by the system to indicate the variable(s) and the conditions, and limitations of the status report desired. For example, TIMM-E can describe the number of hours of training technicians received prior to field duty if the user specifies whether the average number of hours per trainee or the range is desired.

Compare inquiry mode allows the user to make comparisons between the status of different variables. The user is prompted to specify the limits or conditions within which the comparison will be made. For example, if the user wishes to make a comparison between varying amounts of initial training time on technicians and the impact of on-the-job performance, TIMM-E will prompt the user to specify the variable(s) that are to be contrasted on what measure. A statement is provided indicating whether a difference exists, and what the size of the difference is.

Explain inquiry mode provides to the user the probable reasons for one variable being different from another, or from a standard. Explain inquiries prompt the user to identify what is to be explained and specify the limits and conditions within which the variable is to be defined. As in the previous example, the user will ask TIMM-E to explain the higher levels of on-the-job performance of trainees receiving more time for training. The feedback provided to the user by TIMM-E would be in the form of a list of probable reasons for the discrepancy on the performance measure. The reasons listed appear in the order of importance.

Predict inquiry enables the user to see what will happen to a specific variable

interrelates critical factors of training with available data and provides estimates of the impact of alternative decisions. The fourth step is to specify the hardware and software, reflecting the training managers' requirements for the system.

The Rochart and Tracy (1980) approach to the design of DSS's was implemented in the development of two prototype systems for the Army. The next section will describe two systems which are based on extensive analysis of Army training managers' information needs, job/task requirements, and the decision making process.

TWO ARMY TRAINING INFORMATION SYSTEMS

This section will describe the Maintenance Performance System-Organizational (MPS-O) and the Training Information Management Model-Executive (TIMM-E). MPS-O has been designed to support the decision making of first-line maintenance supervisors by tracking the performance of all maintenance activities. By contrast TIMM-E is a prototype DSS designed to provide training managers with a management tool to aid in making evaluation and policy decisions regarding Army training.

Maintenance Performance System Organizational

MPS-O has been designed in response to problems observed in Army maintenance (Harris, 1983). These problems may be categorized in terms of Command Emphasis, Management Information, Management Proficiency, Application of Resources, and Technical Proficiency. Command Emphasis is the importance given to maintenance by Army commanders. They view their operational mission as their prime responsibility and as a result maintenance and maintenance training may be neglected. Management Information is needed so that managers can identify what equipment is maintained, what maintenance tasks are performed, and who performs the maintenance. Management Proficiency refers to the problems which have occurred because managers are not given management training. Application of Resources problems result when maintenance tasks are not assigned appropriately. Technical Proficiency is the skill training and motivation required to perform maintenance tasks successfully. Training on the job is frequently inadequate or non-existent.

The purpose of MPS-O is to help maintenance supervisors diagnose maintenance performance problems, assign appropriate mechanics, identify training requirements, and take actions to meet the requirements. MPS-O covers nine Military Occupational Specialties (MOS) including: Tactical Communications Systems Operator/Mechanic (31V), M60A1/A3 Tank Turret Mechanic (45N), Light Wheeled Vehicle Mechanic (63B), and Fighting Vehicle System Mechanic (63T). MPS-O also covers 11 types of tracked vehicles, 9 types of wheeled vehicles, and 6 types of communications equipment. It can be expanded to include additional MOSs and new equipment. MPS-O is an integrated system comprised of four components: a Management Information System, Action Meetings, Skill Development, and Mechanic Certification. The following sections will provide an overview of each MPS-O component.

Management Information System (MIS). The purpose of the MIS is to provide data on maintenance effectiveness, mechanics' technical skills, and the application of maintenance resources. Supervisors record each maintenance activity and who performs it. The data is stored and processed by a microcomputer. Reports are generated for maintenance management and training management purposes. The maintenance management reports indicate how mechanics are utilized for maintenance activities and the impact of these activities on vehicle readiness. The training management reports track the growth of the mechanics' skills by section, individual, or task.

Action Meetings. Providing managers with information is not a sufficient condition for improved maintenance performance. During Action Meetings managers act on the information they receive in the MIS reports. Action Meetings are also designed to provide feedback to the mechanics and get them directly involved in improving maintenance performance. The Action Meetings consist of the establishment of a formal maintenance work group which meets regularly to review the latest MPS-O reports and assess the unit's maintenance performance. Performance problems are discussed, alternative courses of action are identified, and plans are made to implement changes.

Skill Development. An Individual Skill History is recorded by MPS-O. Mechanics who are new to the unit complete a

if another variable is manipulated. The user is prompted to specify the dimension the manipulation is expected to impact, specifically the variables to be manipulated and size of the expected change. Using the previous example, a user may wish to see the impact levels of on-the-job performance brought about by increasing the amount of training time by 20%. The user will receive a statement of the expected change in the level of on-the-job performance accompanied by an increase of 20% more training time.

Decide inquiry provides to the user a recommended course of action for achieving a specific goal. The user is prompted to specify the goal, what the alternatives are, what values are to be considered, and the level of importance that each factor should be given. A goal may be to improve the on-the-job performance of technicians. Possible actions to be recommended to the user are: increase training time per technician; increase the level of on-the-job training; or, increase the number of job aids available to technicians. And, the values to be considered in selecting an alternative action are, in order of importance: cost of the alternative, length of time to implement, the likelihood of success, and the amount of impact the alternative is expected to have. TIMM-E will then recommend a course of action with the best overall "goodness" of fit. The results of these recommendations are communicated to the user by text, graphics, or both.

In addition to the five modes of inquiry the user has the option of selecting additional analytic procedures from a statistical package. This option enables the user to perform additional analyses on available data bases.

RECOMMENDATIONS

Based on the authors' experience in the development of these two systems, the following recommendations are suggested to assist others in their development efforts.

1. The DSS should be targeted to serve a single population of managers.

2. A comprehensive task/job analysis should form the basis for system design.

3. The DSS should have a self-contained tutorial for managers.

4. The decision model should be based on manager requirements.

5. The analysis used in the decision model should be transparent to the managers.

6. The DSS should have options to help the managers interact with the system. A sample of these options include on-line explanation of errors as opposed to error codes, menus to enable easy access to the complete system, and help segments to guide the user through the decision making process.

SUMMARY

In training, computers have been primarily used for the collection, analysis, and delivery of instructional information. An emerging area of computer-based training are applications designed to help training managers responsible for planning and making decisions about organization-wide training needs. This paper discussed two tools for Army training managers. MPS-O is a Management Information System for first-line supervisors making decisions about individual training requirements. TIMM-E as a DSS, supports top-level training managers working on the broader issues of Army training policy and evaluation.

REFERENCES

Harris, D. (1983). The maintenance performance system-organizational: Description of its components and operation (ANACAPA Sciences Technical Report 465-34). Santa Barbara, CA: ANACAPA Sciences, Inc.

Kriebel, C.H. (1975). The future MIS. In A. Rappaport (Ed.), Information for decision making (pp 25-31). Englewood Cliffs, NJ: Prentice Hall, Inc.

Rochart, J.F., & Tracy, M.E. (1980). Executive information support systems. Boston, MA: Center for Information Systems Research, Sloan School of Management.

Sprague, R.H. & Watson, H.J. (1977a). MIS concepts Part I. In W.C. House (Ed.), Interactive decision oriented data base systems (pp. 7-13).

New York, NY: Mason/Charter Publishers, Inc.

Sprague, R.H. & Watson, H.J. (1977b). MIS concepts Part II. In W.C. House (Ed.), Interactive decision oriented data base systems (pp. 14-23). New York, NY: Mason/Charter Publishers, Inc.

Note: The views, opinions and/or findings contained in this paper are those of the authors and should not be construed as an official Department of the Defense or Department of the Army position, policy, or decision, unless so designated by other official documentation. The Maintenance Performance System-Organizational was developed for the US Army Research Institute by ANACAPA Sciences,Inc. under contract MDA 903-81-C-0032. The Training Information Management Model-Executive was developed for the US Army Research Institute by Human Technology, under contract OPM-26-02.

A Comparative Study of

Computer-Based Instruction vs. Lecture

John C. LeDoux, FBI Academy

Charles J. Stanley, FBI Academy

ABSTRACT

Computer-based instruction was compared to instruction delivered through lecture. New Agent Trainees at the Federal Bureau of Investigation Academy served as subjects. One group received a block of instruction through a highly interactive lecture, a second group through the use of a computer-based instruction module, and a third group through both methods. The students using micro-computers in the learning experience scored significantly higher on an achievement test, required less time to complete the lesson, and rated their instructional method higher than did those students who attended only the lecture.

INTRODUCTION

A synthesis of the information available on adult education and computer-based instruction[1] suggests that the use of computers for instruction, if employed properly, could provide at least three primary benefits: 1) achievement of higher levels of learning since students will be actively involved in the learning process and receive immediate feedback to correct any substantive misconceptions concerning the subject matter; 2) increased learner satisfaction and motivation since the student will feel a greater sense of control over the learning experience; 3) the ability to cover the subject matter in less time.

The purpose of this study was to compare computer-based instruction (CBI) to instruction delivered through lecture. The primary contrast was student short-term learning. Student ratings and the amount of time necessary to complete the lesson served as the basis for other comparisons.

METHODOLOGY

Subjects

New Agent Trainees in two sections at the FBI Academy, Quantico, Virginia served as subjects. The men and women ranged in age from 23 to 35 and each possessed at least a baccalaureate degree. Students were randomly assigned to one of three groups within each section. The specific treatment (method of instruction) received by each group also was randomly determined. Each group consisted of ten students except for Section 2 "Lecture" group which, for administrative reasons, consisted of fourteen students.

Treatments

One treatment group, indentified as "Lecture", received a block of instruction through a highly interactive lecture. After review of pertinent legal statutes, the instructor constantly posed questions as an investigative scenario was discussed. The instructor would ask, for example, "What's your next logical step?" Therefore, the term "lecture" is used in this case to indicate an instructor interacting with the class rather than an instructor talking to a passive group of note takers. The time allotted for this instruction was 50 minutes.

A second group, indentified as "CBI", received the block of instruction through the use of computers. The

CBI treatment was developed in conjunction with the lecturer who served as the subject matter expert. The lesson module was designed using the McGraw-Hill Interactive Authoring System. Students worked individually through the lesson on IBM PC microcomputers with monochrome monitors. The fifty-eight screen lesson was based on exactly the same investigative scenario employed in the lecture.

A third group, indentified as "Lecture/CBI", received instruction through both treatment methods, first through the lecture and then through CBI.

Test Instrument

To test the students' knowledge of the subject matter, a six item multiple choice test was developed also in conjunction with the lecturer. The questions were based on the instructional objectives of the class and were reviewed by other subject matter experts to ascertain content (face) validity. The information necessary to achieve the instructional objectives was equally emphasized in both the lecture and the CBI module.

Student Rating Form

A nine question instrument (see Figure 1) utilizing an anchored seven-point Likert scale was developed to obtain student perceptions concerning their particular instructional treatment. The first seven questions were anchored on one end of the continuum with "Not at all Descriptive" and on the other end with "Highly Descriptive". The anchor points for the final two questions were labeled "Poor" and "Excellent".

Data Collection

All students completed the knowledge test during the class prior to participating in their respective instructional treatment. This served as a pre-test to determine if a difference in subject matter knowledge existed between groups.

During the next session both the "Lecture" and the "Lecture/CBI" groups from a particular section attended the lecture while the "CBI" group was participating in the computer-based instruction. The "Lecture/CBI" group also went through the computer-based instruction at a later time.

Figure 1. Student Rating Form Items

1. The subject matter was thoroughly covered.
2. The class was an enjoyable experience.
3. The class gave me individual feedback to the responses I provided.
4. The class developed my ability to work an actual bribery case.
5. I felt I had control of the pace of the class.
6. The class was challenging.
7. I would recommend this type of instruction to other students.
8. The overall quality of instruction.
9. The overall quality of course content.

An observer kept record of the time required for students to complete the CBI lesson. The knowledge test and the student rating form were administered during the next class on a subsequent day.

RESULTS

Achievement

The knowledge test was scored using a Scan-Tron 888 test scoring machine. Analysis of variance was used in the statistical analysis of both sets of knowledge test scores. The preliminary test results were used to determine if the groups differed prior to undergoing the instructional treatment. There were no significant differences (.05 level of confidence) among the groups in prior knowledge of the subject matter.

Significant F-ratios were obtained for the post-test knowledge data indicating that there was a difference among the treatment groups' test scores within each section as a result of the treatment. The derived test score means and F-ratios may be viewed on Table 1. A Sheffe procedure for post-hoc comparisons was used to determine which specific groups differed significantly from one another. Both the "CBI" and the "Lecture/CBI" groups' mean scores individually were significantly higher than the respective "Lecture" group's mean score (.05 confidence level). The "CBI" and the "Lecture/CBI" groups in either section did not differ significantly from one another. Accordingly, the remaining data analysis will focus on the comparison of the "CBI" and the

"Lecture" groups.

Student Rating of Instructional Method

Differences in perceptions of instructional method quality were examined through comparison of the "Lecture" and "CBI" group members' responses on the student rating forms. A one tailed t-test (.05 confidence level) was used for the statistical analysis.

There were numerous significant differences between the "Lecture" and the "CBI" groups. These are illustrated on Table 2. Section 1's "CBI" group rated their method of instruction significantly higher than the "Lecture" group in all nine areas assessed by the evaluation instrument. Ratings of the "CBI" group in Section 2 were significantly higher than their "Lecture" counterparts on five of the nine items. Higher ratings were awarded by both "CBI" groups, as compared to the "Lecture" groups, in terms of the lesson being an enjoyable experience, providing individual feedback, having control of class pace, being challenging, and whether students would recommend this type of learning experience to other students.

Comparisons were made also between the same treatment groups in the two different sections again using a one tailed t-test. Table 3 summarizes these findings. The Section 2 "Lecture" group, as compared to the Section 1 "Lecture" group, rated their treatment higher in terms of the subject matter being thoroughly covered, the class being a challenge, recommending that type of instruction, the overall quality of instruction, and the overall quality of the course content. Interestingly, the students in the Section 2 "Lecture" group collectively demonstrated slightly less achievement

Table 1. Post-treatment Knowledge Test Score Means and Between Groups F-Ratios

Section	Group	Mean	F
1	Lecture	66.693	
	Lecture/CBI	95.002	21.257*
	CBI	86.668	
2	Lecture	64.285	
	Lecture/CBI	90.002	16.944**
	CBI	93.334	

* p<.05 (2, 27 df)
** p<.05 (2, 31 df)

Table 2. Student Ratings Means by Section with Between Groups t Values

	Section 1			Section 2		
Item	Lecture (n=10) Mean	CBI (n=10) Mean	t	Lecture (n=14) Mean	CBI (n=10) Mean	t
1	4.80	5.60	4.56*	5.57	5.50	-0.15
2	4.60	6.60	3.42*	5.30	6.40	3.47**
3	3.90	6.00	2.51*	5.00	6.10	2.60**
4	4.00	5.70	2.04*	5.14	5.40	0.76
5	4.27	6.40	3.24*	5.00	6.90	5.58**
6	4.00	5.88	3.04*	5.07	6.10	3.13**
7	4.40	6.40	3.81*	5.38	6.20	2.18**
8	4.70	6.00	2.11*	5.57	6.10	0.91
9	4.40	5.90	2.27*	5.64	6.00	0.98

*p<.05, 18 df
**p<.05, 22 df

Table 3. Student Rating Means with Between Section t Values

	Lecture			CBI		
	Section 1 (n=10)	Section 2 (n=14)		Section 1 (n=10)	Section 2 (n=10)	
Item	Mean	Mean	t	Mean	Mean	t
1	4.80	5.57	1.91*	5.60	5.50	-0.19
2	4.60	5.30	1.44	6.60	6.40	-0.73
3	3.90	5.00	1.62	6.00	6.10	0.18
4	4.00	5.14	1.36	5.70	5.40	-0.57
5	4.27	5.00	1.02	6.40	6.90	2.06**
6	4.00	5.07	2.12*	5.88	6.10	0.81
7	4.40	5.38	1.91*	6.40	6.20	-0.60
8	4.70	5.57	2.04*	6.00	6.10	0.26
9	4.40	5.64	2.13*	5.90	6.00	0.26

*p<.05, 22 df
**p<.05, 18 df

although the difference was not statistically significant. The ratings of the "CBI" groups in both sections differed only in terms of the students' feeling of control of the class pace indicating more consistency in student ratings with the computer-based instruction.

Time Requirements for Lesson Completion

Table 4 contains summary data concerning the amount of time necessary for completion of the CBI lesson. Half of the students in the "CBI" group finished in twenty-five minutes or less. This amounted to a fifty percent time savings when compared to the fifty minute interactive lecture. The slowest student in one "CBI" group took thirty-five minutes, a thirty percent time savings. The slowest student in the other "CBI" group took forty minutes, a twenty percent time savings. These two students went through the lesson two times and took copious notes.

CONCLUSIONS

On the basis of the results reported above, computer-based instruction is more effective in achieving student short-term learning. Students exposed to the subject matter only through CBI scored significantly higher on a knowledge test than did students who were exposed to the same subject matter only through lecture. As one might expect, students who were exposed to the subject matter through both instructional systems also scored significantly higher than did those who only attended the lecture. However, neither "Lecture/CBI" groups scored significantly higher than their "CBI" counterparts.

Of particular note, the lecture was one in which the instructor maintained a high level of interaction with the students, not one in which a "one-way" communication strategy was employed. Thus, we believe that the CBI approach was compared against particularly stringent competition.

Table 4. Time in Minutes Taken to Complete the CBI Lesson by Section

	CBI		Lecture/CBI	
Section	1	2	1	2
Completion Order				
1st Finished	15	19	16	10
5th Finished	23	25	18	17
Last Finished	35	40	25	20

Students in the "CBI" groups tended to rate their method of instruction significantly higher particularly in terms of the lesson being an enjoyable experience, providing individual feedback, having control of class pace, being challenging, and willingness to recommend this type of learning experience to other students. Furthermore, there was more consistency by far in the ratings of two "CBI" groups than in the ratings of the two "Lecture/CBI" groups.

Some resistance to CBI expressed by several teaching faculty members with whom we are acquainted stems from the belief that no piece of machinery can equal a good instructor when he/she is "on". While we find some potential merit in the position, the present study suggests that consistency in delivery must be considered when comparing various instructional forms. As noted above, there was little difference in the student ratings from one CBI presentation to the next. However, this study seems to have captured a situation in which the instructor and his lecture were perceived as "good" by one group of students and as "excellent" by a similar group.

Finally, the use of computer-based instruction was more efficient in terms of the time required to deliver the subject matter. Students who were exposed to the subject matter through CBI, therefore, demonstrated higher levels of achievement, rated their learning experience higher, and learned the information more quickly.

The results indicate that computer-based instruction has the potential of being a powerful and efficient method of training. However, this conclusion must be presented with a caveat. There must, of course, be a match between the instructional goal of a lesson and the capabilities of the instructional delivery method. The scope of this paper does not permit a discussion of all the considerations when constructing creditable CBI. This lesson centered on a progressive problem-solving approach similar to the decision-making actually faced by a Special Agent of the FBI when investigating an alleged federal violation. CBI, therefore, was a logical instructional strategy to employ. The student could be afforded the experience of "investigating a real case" without running the risk of committing errors which would damage the prosecution of a case.

Footnotes

[1] See, for example, Dean, C. & Whitlock, Q. A Handbook of Computer-based Training. London: Nichols Publishing Co. (1983); Knowles. M. S. The Modern Practice of Adult Education. New York: Association Press (1970).

****

John C. LeDoux is an FBI Special Agent assigned to the FBI Academy in Quantico, Virginia. He holds a doctorate in Adult and Vocational Education from Auburn University.

Charles J. Stanley is an Instructional Systems Specialist at the FBI Academy. He is a doctoral candidate in the Higher Education program at the University of Virginia.

PROJECT U-TRAIN CAL

ENTERING THE HOME STRETCH

Major Harry M. Nash

Canadian Forces Training System Headquarters

ABSTRACT

Project U-TRAIN CAL was initiated for the purpose of: developing a methodology for Computer Assisted Learning (CAL) in the Canadian Armed Forces, defining optimum configurations of CAL systems, carrying out a cost/benefit analysis to support a decision on future implementation of CAL, and evaluating the NATAL-2 courseware authoring language. The project has developed all courseware to be used in this investigation and is currently involved in delivery to students and data gathering. Preliminary data has already been submitted, and many lessons have been learned in the areas of courseware authoring and development techniques. These lessons are discussed in this paper. For a more detailed statement of the earlier progress of this project, several references are available. (Bélanger and Sapp, 1983; Bélanger, Sapp and Drake, 1983)

INTRODUCTION

Project U-TRAIN CAL is a test of the concepts of self-pacing using computer technology. The project uses the NATAL-2 courseware authoring language, developed by Canada's National Research Council and implemented by Honeywell Information Systems on their Level 6 minicomputer. Courseware is delivered via 12 intelligent multi-media workstations using touch-sensitive slide and CRT screens and random access audio devices.

The project has developed six technical courses for evaluation. Five of these relate to the maintenance of the C-130 Hercules aircraft and one is a digital computer fundamentals course for electronics technicians. All of these courses are being delivered to students at this time.

Project staff intend to deliver each of these courses to a sample of 40 students per course during the first six months of 1985. The remainder of the year will be taken up with field validation and data gathering.

APPROACH TO COURSEWARE DEVELOPMENT

The Canadian Forces supports the concept that the best people to train students are those who have spent their time as a job incumbent and have achieved the rank and expertise of a Subject Matter Expert (SME). In the case of our six technical maintenance courses, the SME must have:

- maintained the equipment on the line,
- reached the expert level, and
- have instructed in the subject for at lease one year.

Upon assignment to the CAL project, the SME is provided with training in the concepts of Instructional Systems Design (ISD) and the NATAL-2 authoring language. All of the required training can take as little as three weeks, but is largely dependent on the ISD method chosen.

The SME is then tasked with carrying out a new analysis of the subject, with the guidance of an expert in training development and ISD procedures. Only after all objectives had been clearly defined as a result of a job and task analysis, does the SME receive permission to prepare and program the course. Failure to comply with this requirement has produced poor courseware in the past.

APPROACH TO COURSEWARE DELIVERY

The project takes a multi-media approach to courseware delivery. Specifications for the student workstations included a requirement for two touch-sensitive screens, one for 35mm slide projection, and one CRT; terminals that could present high-resolution colour graphics; and a random access audio unit.

These workstations were built under contract and are now being used to present courseware. They were designed to provide the author with the tools to cause the student to hear, see, and touch as much as possible, therefore, to aid learning. In our investigation, an attempt is being made to determine the impact of single or multiple sensory exploitation so that optimum configurations can be recommended at project completion for future development.

TYPE OF TRAINING

Because of the considerable front-end costs of CAL, the courseware that was developed was selected from subjects that dealt with equipment and concepts that would not need extensive change or replacement in the near future. Since our C-130 Hercules aircraft, in its current configuration, has been assigned a life expectancy of another 25 years, it was felt that courseware designed for Aero-Engine and Instrument/Electrical technicians who work on this aircraft would be used extensively. Since the CF had a backlog of approximately 1600 personnel awaiting training in digital computer fundamentals in 1983, it was decided that CAL would be an excellent delivery vehicle for this type of training as well.

APPROACH TO EXTERNAL EVALUATION

The project will be finished its courseware delivery phase in mid-year and will concentrate its efforts on field validation. Students are provided with a pre- and post-test on the subject at the learning centre. Valiation will consist of a test of the graduate at the worksite and interviews with both the graduate and his/her supervisor. We intend to perform validation with each student between four and six months after graduation.

COMPARISONS OF ASSUMPTIONS AND FINDINGS TO DATE

The project team began with a number of assumptions, some of which have been reinforced, and some of which discarded. We are now at the point where limited delivery/validation has been carried out. While no decisions can be made using such a limited sample, trends are evident. Following is a series of assumptions that were made at the beginning of the project, accompanied by our findings to date. Evidence will be professionally reported at project end, however, this is the up-to-date evidence we have at hand for our configuration.

ASSUMPTION: The technician Subject Matter Expert can be trained to be a CAL author programmer.

FINDING: TRUE. The SME technicians adapted readily to ISD concepts and all were able to program using the NATAL-2 language, without exception. During the first three or four months, SMEs kept to the programming of presentation material, but natural curiosity and interest caused them to experiment with the language and develop their own control procedures. Minimum mastery levels have therefore been achieved within two months of joining the project.

ASSUMPTION: Training the SME in NATAL-2 and ISD could be done quickly and easily.

FINDING: TRUE. A three week training period may be all that is required. the ISD resource person, available to the SME, was used frequently at the beginning, but after two to three months of experience, he was called upon only at significant milestones of courseware development. It is foreseen that, should result analysis indicate that different instructional strategies be required, the ISD resource person would be required on a continuing basis.

ASSUMPTION: The CAL courseware development centre could be staffed with 1 systems operator, 1 training developer, 1 language consultant, and the remainder SMEs.

FINDING: TRUE. The need for a language consultant is rare, but this person was also the Project Leader, and therefore was required as administrator.

ASSUMPTION: All SMEs could develop courseware on a part-time basis and still carry a normal classroom instruction load.

FINDING: FALSE. The average instructor on the project had a load that allowed him to devote approximately 20% of his time to courseware development. The problem was not so much the amount of time that he could devote as much as the way that this time was broken up. The author would sit down and familiarize himself with the development task, perform some work, and then return to his primary duties. The division of time was such that the familiarization took up as much, if not more, than the development time. This resulted in frustration on the part of the SMEs, and courseware that lacked continuity. A large percentage of SMEs opted out of the program. The project was severely hampered because this was one of the foundations upon which it was initiated. As a result, courseware revision was required and the project completion date was extended considerably. (Bélanger, Sapp and Drake, 1983)

ASSUMPTION: Conceptual branching is essential in courseware development.

FINDING: PROBABLY FALSE. Branching has been applied only in very limited circumstances. There is no evidence to date that shows that different interpretations of lessons have had an effect on student comprehension in our packages.

ASSUMPTION: Development factors for CAL range from 100 to 200:1.

FINDING: FALSE. All courseware used in the project has been developed from a proper task analysis, through to performance and enabling objectives. Development factors range from 90 to 120:1, including this analysis.

ASSUMPTION: CAL will inevitably result in a reduction of training time if ISD procedures are used.

FINDING: TRUE. However, the specific reason for average course reduction time is clouded through measurement difficulties. CAL courses in our approach are self-paced and receive a vigorous ISD application. In addition, one can entertain that in certain courses, presentation of some aspects like drill and practice sessions provide for efficiencies because of the technology involved in presenting the materials. It is our opinion that each courseware application requires a specific analysis to determine what amounts of training time will be 'saved' over conventional instruction.

REFERENCES

Bélanger, J.R., Sapp, R.D., Drake, R.D. (Oct 1983), Project U-TRAIN/CAL: From Military Instructors to NATAL Programmers. Proceedings Fourth Canadian Symposium on Instructional Technology, Winnipeg, Manitoba

Bélanger, J.R., Sapp, R.D. (Aug 1982). Project U-TRAIN/CAL: NATAL as an Authoring Language, Journal of the Association of Education & Training Technology, Vol 21 No. 3

EFFECTIVE MANAGEMENT OF CAI PROJECTS

Mary Jac Reed
George Reed
University of Delaware
Office of Computer-Based Instruction

ABSTRACT

Educational software development in the past has been a one person operation, staffed by the interested professor or the computer-literate teacher. Some notable computer programs have been produced in this manner.

With a large installed hardware base, the software market is now growing rapidly. To meet production schedules within reasonable timeframes, while still producing quality software products, the software development process should be built around teams of designers, programmers, and content experts, and directed by able and knowledgeable managers. The role of the traditional manager and the skills that person must possess to successfully perform the tasks will provide the framework for applying effective management to CAI projects.

INTRODUCTION

Educational software development has been a cottage industry for the past ten years. Usually a one-person operation, it has grown up around colleges and grade schools alike, staffed by the interested professor, the computer-literate teacher, or the computer fancier. Some notable computer programs have been produced in this manner, together with a large quantity of less than useful educational materials.

As is often the case with technological advances, the computer hardware market is years ahead of software production. With a large installed hardware base, the software market is now growing rapidly. As a consequence, the computer software producers now find themselves beset with requests for predictable timelines, accurate cost projections, and rapid production schedules.

Software development no longer belongs solely to the educator-turned-programmer. The pressures of the software market are forcing developers to apply traditional managerial theory to insure the smooth functioning of computer assisted instructional (CAI) development projects.

This paper will examine a software development model for computer assisted instruction, and the role that a manager should play in the process.

FUNCTIONS OF MANAGERS

The traditional role of managers can be divided into four main areas (Fig.1). During the PLANNING phase, managers lay the groundwork for a project, identifying objectives and goals. The project's course of action is charted.

The next step is ORGANIZING. An equitable chain of command is established and a division of work is suggested. A large part of this organization phase is staffing, which means finding qualified people and matching them to the available position.

DIRECTING involves motivating the work force, providing leadership, and instilling employees with loyal allegiance to their project. Good managers use the human resources they have at hand to accomplish the goals of the project without jeopardizing the growth and welfare of the team members.

When managers are CONTROLLING, they are keeping track of progress, comparing present timelines with projected target dates. They are evaluating the organizational activities and the members of the project, and deciding what needs to change in order to attain the project goals.

Managers have traditionally been evaluated on how effectively and efficiently they accomplish tasks and manage their project. Managers are asked to achieve the desired results, while minimizing the problems and maintaining order.

THE CAI PROJECT MANAGER

The CAI manager shares many of the same functions with the traditional manager, even though the application is towards a new field and product.

Planning

Planning is still the most important task that befalls the manager. In the earliest stages, the manager may be the only team member on a project. It is that much more important that the manager be familiar

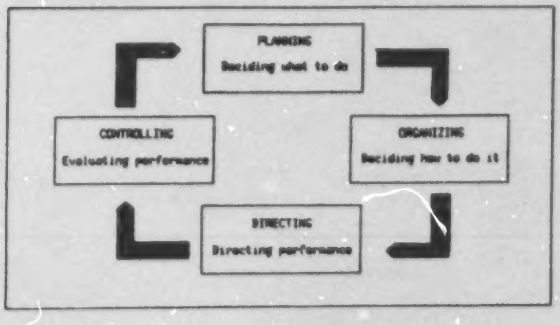


FIGURE 1. The Traditional Role of the Manager

with the processes and the pitfalls that often doom a CAI project, such as understaffing, unclear or overly ambitious goals, and timelines that are much too short.

Ideally, if a manager is planning a project from its conception, a needs assessment is completed. A content area is targeted, one that has not yet been successfully implemented through CAI, and one in which staff have some proficiency. A task analysis is helpful in determining what material within a subject area is most adaptable to a CAI approach. Other media already used in this field can be identified and assessed for their effectiveness, avoiding unnecessary duplication of effort. The extent of the project should be defined, to avoid too large a project and to insure that the CAI component is employed only where it is needed.

In reality, most CAI projects are given to a manager at some later stage in the process. The content has been selected, the extent of the course has been prescribed, and most likely a deadline has been set.

No matter when a manager assumes the leadership of a project, decision-making begins immediately. The managers must determine what equipment is appropriate for development and for delivery. A good working environment, both physically and in terms of software tools, for designers and programmers, must be established, if serious production output is expected.

Organizing

Initially, the extent of the project is determined, the equipment is selected, the developmental tools are either marked for purchase, if they exist, or marked for development, if they do not. Tentative staffing needs are listed, along with estimated time commitments. Development sites are readied, and development models are detailed.

Several time management aids are available to managers estimating the time-line for a project. One useful tool is a Gantt chart (Fig.2). This chart depicts the relationship between the work planned and completed on one axis and the time elapsed on the other axis. With so many team members and with such a complicated process of production and revision, it is

BEST COPY AVAILABLE

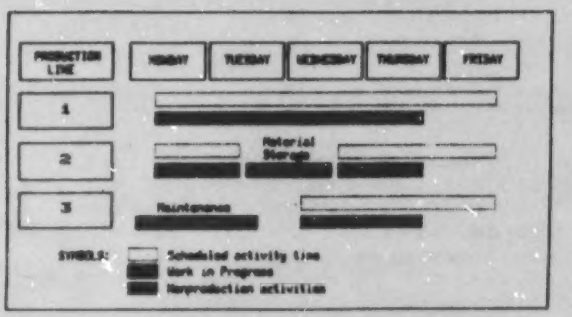


FIGURE 2. A GANTT Chart

important to be aware of how many tasks each member of the project is being asked to accomplish at one time. The modified Gantt chart (Fig. 3) identifies the overlap that occurs when several lessons are being produced simultaneously by a single project.

When staff are selected, preference should be given for prior software development experience AND expertise in the content area. Development proceeds much more quickly if production team members are familiar with the subject matter (Garton, et al,1984a). If this is true, then designers can present ideas for instruction presentation, and fewer content errors will be introduced during lesson development.

An initial staff should be selected, consisting of a content area expert, a programmer/analyst and a lead designer. These staff persons will have responsibility and authority over all other content experts, designers, and programmers who might be added to the project at later dates. The subject matter expert should be appointed as the lead content person, with the power of arbitration, to decide any content controversy that may develop among the experts. Together with the manager, this team core will detail the steps and stages in development, refine the timeline, and recruit the staff. Experience is the best estimator of the time and staff required for a large CAI project, but it is useful to see what other projects have needed (Garton, et al, 1984b).

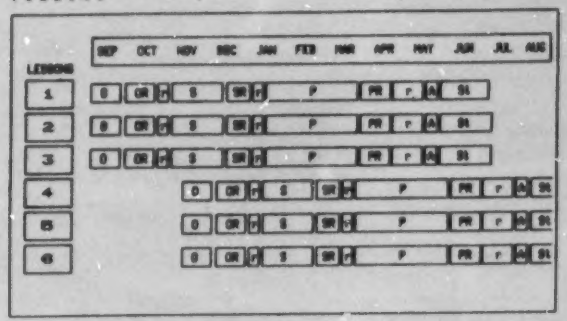
This advanced organizing, so necessary at the beginning of a project, should continue in a dynamic way throughout the project, subjected to constant and routine revisions, as other factors enter into the project and alter the product, the staff, and the timeline.

Directing

The manager now has created another level of management within this project (Fig.4). There is a programming coordinator, a design coordinator, and a subject matter expert, each in charge of a specific phase of the development, with responsibility and authority for their area and for the people working under them.

The manager's role is to define this team differentiation clearly and concisely for the team members. The subject matter expert will head a team of other content experts and determine exactly the parts of the content each person will contribute. An overall instructional strategy for all the lessons will be planned at this stage.

The designer would initially plan a general guideline for the design of the



KEY	ACTIVITY	TIME
O	CONTENT OUTLINE	(3 weeks)
OR	OUTLINE REVIEW	(3 weeks)
r	OUTLINE REVISION	(1 week)
S	SCRIPTING	(5 weeks)
SR	SCRIPT REVIEW	(2 weeks)
r	SCRIPT REVISION	(1 week)
P	PROGRAMMING	(10 weeks)
PR	PROGRAM REVIEW	(3 weeks)
r	PROGRAM REVISION	(3 weeks)
A	LESSON APPROVED	(1 week)
St	STUDENT TESTING	(5 weeks)
	TOTAL:	37 weeks

ASSUMPTIONS

1. Team consisted of: 3 content experts 3 designers 6 programmers
2. At most, 3 lessons could be outlined at one time.
3. Same for scripting.
4. 1 programmer could work on one new lesson and revise one old one at one time.

FIGURE 3. A Modified GANTT Chart

BEST COPY AVAILABLE

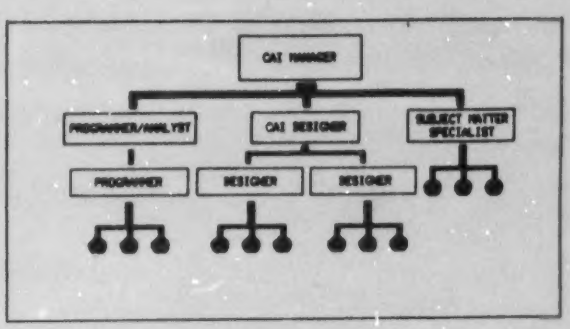


FIGURE 4. Organizational Chart of a CAI Project

and organizing the flow of individual lessons (Reed, et al, 1983). The designers will watch over the package to insure for consistency of presentation, style, display layout, and the mechanical operation of the lessons. The programmer/ analyst can begin immediately to develop tools that the team has targeted, such as a graphics editor or database router. The analyst is responsible for the background programming, that will enable all lessons to operate in a consistent manner, with the least duplication of code among other programmers. This task might require establishing a library of coding routines, available to team members from a central directory.

The manager is responsible for communications. The three coordinators must inform the manager of decisions, problems, and progress. In return the manager must keep each coordinator up-to-date about developments within the project. Scheduled meetings of the coordinators facilitate this task and allow the exchange of ideas among the three component groups. Daily contact with team members and extensive documentation of the project's progress is a vital part of managing a CAI project, most effectively and efficiently carried out through electronic mail (Stabosz & McGivern, 1983).

Controlling

Controlling is concerned with the evaluation of organizational activities as they relate to stated goals and plans. This final function of a manager is often times forgotten in CAI lesson development. Time is often at a premium, and the philosophy of 'if it works, don't fix it' can prevail. But to insure quality software, it is crucial not to sidestep the evaluation process of the product.

Quality assurance should be built into various stages of the development, with ample opportunity for input from all team members at many points within the process (Seiler, 1978). Formal reviews at the content outline stage, by the team and by colleagues of the content experts, allow incorporation of good ideas before too much effort has gone into the product. A review of the content after it has been adapted to the computer screen, but before it has been programmed, is recommended. Material is being presented on a new media and some team members are probably unfamiliar with the potential of the computer. A review at this time stimulates new ideas, seeing the material, display by display, adapted for the computer screen. The novice developer may be inspired towards new frontiers. Several reviews should be scheduled, once the lesson is programmed, one as an internal review for team members, another review for colleagues of the content experts, and finally a review when the lesson is tested by the end-users.

One result of such an extensive reviewing process is an even more extensive revision cycle. A manager who is aware of this effect can plan adequate time for revisions into the timeline. However, there may still be a need for enforcing a limit to revisions, especially ones that depart dramatically from previously approved drafts (Reed & Smith, 1983).

THE CAI PEOPLE MANAGER

In addition to managerial functions that allow managers to do things in a systematic way, successful managers must possess certain managerial skills to perform well in their role. These behavioral skills can be classified into three main types: interpersonal, informational,

and decisional.

Information processing and decision making are obvious parts of a management position. Interpersonal skills may not be as apparent, but are the most crucial for a successful manager to develop. It is important for the CAI manager to manage both the product and the people producing the product.

An effective method towards accomplishing both goals is to establish a strong team. The team interacts to discuss and distribute ideas about the product, and can provide ideas, support, and motivation to each other when a given task seems formidable. The most effective means for achieving a good esprit de corps is through an unambiguous chain of command where one subordinate receives direction from only one supervisor. More subtly, a mixture of both oral and written communications creates a more personal working environment and unifies the efforts of the team.

It is important to understand what motivates the programmers and designers on a CAI project. They will be expected to put in long hours and operate at a high level of creativity. Money is only one factor that provides job satisfaction, and is not always the primary factor. A challenging task and a voice in decision making are two powerful factors to consider when attempting to motivate the team. The manager should retain final responsibility but give the team members enough authority to complete the task successfully. Each team member should feel assured that they have the ability and initiative to suggest new ideas, and that they have the support of the team and management to follow through with their implementation.

The team should feel a sense of pride in their accomplishments of having produced a quality software product. The programmers and designers expend much creative energy to achieve the goals of software production. They cannot feel that they are automatons, producing an assembly line product. A royalty agreement that divide any profits from the marketed product equitably among the team members is an incentive measure to insure that team members will benefit.

Product timelines can be distracting. The challenge to finish on time may prevent the manager from fulfilling an equally important function -- seeing that the team members develop personally and professionally while involved in this project. A really successful work team is built upon the interpersonal interactions of its members.

CONCLUSION

The computer has revolutionized the way we communicate, educate, and conduct business. Yet the new field of software development is similar to other more traditional production operations. People make up the CAI team, working behind the computers, and developing the educational lessons. The way people work and interact with one another does not change with the product produced. The manager's primary function is to work with people to make workers happy, productive, and satisfied.

If CAI developers are to keep pace with the demands of this fast moving industry, then traditional managerial strategies should be adapted to the management of CAI projects and people.

BIBLIOGRAPHY

Garton, R.A., Reed, M.J., Reed, G.A., & Stevens, E.V., "Developing a CBI Course: The Process." In 1984 Conference Proceedings ADCIS/CBI: Courseware Transportability, pp.143-148, Columbus, Ohio: ADCIS, May 14-17, 1984.

Garton, R.A., Reed, M.J., Reed, G.A., & Stevens, E.V., "Programming a CBI Course: A Case Study." In 1984 Conference Proceedings ADCIS/CBI: Courseware Transportability, pp.149-155, Columbus, Ohio: ADCIS, May 14-17, 1984.

Reed, G.A., Ashley, M., & Reed, M.J., "Planning Large CAI Packages." Alberta Printout. Vol.4, No. 3, pp.20-23, September, 1983.

Reed, M.J. and Smith, L., "Developing Large CAI Packages." CALICO Journal. Vol. 1, No. 3. pp. 13-16, December, 1983.

Seiler, B.A., "Applying a Systematic Approach to CBE Development in an Academic Environment." In 1981 Conference Proceedings ADCIS/CBI: Frontiers of Thought, pp.305-310, Atlanta, Georgia, ADCIS, 2-5 March, 1981.

Stabosz, R. & McGivern, L. "Bureaucracy in CBI Environments: Developing a Policy for Student Authoring." In 1983 Conference Proceedings ADCIS/CBI: Denver, Colorado, ADCIS, April, 1983.

Szilagyi, A.D. "Management and Performance, Second Edition." Dallas: Scott, Foresman and Co., 1984.

The Innovation Diffusion Process in Education - Will Computer-Based Instruction Survive?

Richard L. Reinhart

Microtel Learning Services

ABSTRACT:
Introducing innovation to our educational system has always been a slow, uncertain process. Computer-based instruction is the latest of a long line of technologies to face this. There have been many studies of innovation diffusion that reveal a cross-field (education, entertainment, business) similarity. There are five stages: 1. Awareness; 2. Interest; 3. Evaluation; 4. Trial; 5. Adoption. People and institutions can also be classified as follows: 1. Innovators; 2. Early Adopters; 3. Early Majority; 4. Late Majority; 5. Laggards. Computer-based instruction is basically situated in the Evaluation stage with some limited overrun in the Trial stage. There are several factors that can influence the successful adoption of an innovation outside of its own characteristics. These include timing, advertisement, perceived need and cultural awareness.

Education has always been one of the slowest fields to adopt an innovation. It does not return a good profit and generally is introduced to "spin-off" technology which has succeeded elsewhere. Some of the reasons education is slow is: 1. Low funding; 2. Relatively little invested in Research and Development; 3. Difficult to measure outcomes and evaluate success. We must recognize these factors and how they may affect computer-based instruction as the field of education has witnessed much innovation failure and a successful adoption rate that often "lags 25 years behind the best practice".

Introducing innovation to the educational system has always been a slow, risky process. It took thirty years from its introduction for early educators to view the chalkboard as essential to teaching (Anderson, 1962). How long will computer-based instruction take? Will it even "make it" as a wide-spread teaching methodology in our schools? Will our educational system be changed by computer-based instruction or will computer-based instruction be changed by our educational system? These are questions that will someday be answered to determine if this latest innovation is successful in this field. Komoski (1971) pointed out, "The specific application of technologies tend to conform to, even end up reinforcing, existing educational forms and practices, while at the same time their more general informal impacts on education are making it essential that new educational forms and practices be developed."

The process of introducing, accepting or rejecting a new idea, technology or product is known as innovation diffusion. If we study this process, we learn that systems or ideas that we feel are "obviously superior or necessary" may not succeed on such qualities alone. Sometimes elements that have no relation to the innovation characteristics are key factors that

influence success or failure. Examining this process may help us understand the diffusion of computer-based instruction in education and even influence its success.

It would be fantastic if we could look at the characteristics of a new innovation and immediately determine whether or not it will be successful. There would be millions of research, development and promotional dollars saved. Such a formula does not exist but there is evidence that there are certain conditions that favor the success of an innovation.

Though these conditions may generally apply for innovation diffusion through several fields (i.e. education, entertainment, agriculture, etc.), there are unique characteristics of each field that must be considered. For example, when television was introduced as an entertainment medium, it diffused rapidly whereas this was a painfully slow process in schools where its success may still be debated. Still, the similarities of the innovation diffusion process among fields were largely unrecognized. Everett Rogers (1962) wrote a book that compiled many different studies across several fields that rehashed much of the same information. He marvelled how innovation studies never diffused.

Throughout the many studies that Rogers compiled, five basic stages were determined to be the most common in innovation diffusion. They are:

1. Awareness
2. Interest
3. Evaluation
4. Trial
5. Adoption

Some studies placed these stages in a slightly different order and some added what seemed to be sub-stages, but they all shared this basic process.

The Awareness stage is a point where people somehow have gained the knowledge that the innovation exists. At this point, people were not seeking out this idea but "came across it" either through conversation, advertisement or a popular periodical. Computers have been around a long time but advertisement of educational software is relatively recent. The Awareness stage for computer-based instruction has already passed as few, if any, educators have not been exposed to information of this nature. For most educators, the idea of expert systems (an automated method of storing an expert's knowledge in an interactive format usually intended for problem solving situations) would be in the Awareness stage.

The Interest stage is entered when this "aware" person or group decides to take positive action to acquire more information about the innovation. At this point, these people may seek out a professional journal that expounds on the topic, ask others that are informed, or view a demonstration. Most educators have passed this stage for computer-based instruction though some still lag behind.

The Evaluation stage is when the questions are asked. An educator would compare computer-based instruction (or any other innovation) with present methods of instruction. They should then ask if the advantages of the innovation outweigh the disadvantages. Most of the educational field is in this stage.

The Trial stage is underway when the innovation is acquired and used in a limited fashion. The emphasis here is on limited use. An example of this would be found in the introduction of hybrid seed corn in the 1940s. Farmers would plant only one acre and determine the results instead of committing all their land to this innovation.

In the educational situation, this could mean applying some innovation to one school in a district or one classroom in a school. Today many schools are experimenting with computers in educational situations but relatively few use them to deliver instruction.

The Adoption stage is entered when it is decided to fully integrate this innovation. This could range from planting the entire farm with hybrid seed to implementing computer-based instruction throughout a school district. In the field of education, the Adoption stage may be hard to define because the field is so large and diverse that it is hard to differentiate between successful adoption and what may be just limited general use.

As the stages of innovation adoption are compiled from past research, Rogers also classified the people who adopt an innovation. They are basically classified by the relative time it takes them to adopt the innovation. When group size versus adoption time is plotted, it forms a classic "bell" curve.

The first groups to adopt an innovation are called the innovators. They make up approximately 2 - 3% of the adoption population and are often considered the radicals of the field. They are usually not well respected and do not have warm relationships with neighbors who often think that they are "crazy". As a contrast, they are followed by an approximately 13 - 14% group called the early adopters. These people, who are generally well-liked and respected by the community have also been labelled as "opinion leaders".

Innovators and early adopters are generally better educated, more aware of events outside their local community, and wealthier than the slower adopters. This wealth, of course, makes it easier to take the risk of an early innovation adoption. The opinion leaders have an extra bonus of high social status and their successful use of an innovation is a necessity before the majority of the local population will accept the innovation.

The next group is known as the early majority and make up approximatley 34% of this population and may deliberate for some time before following the example set by early adopters. They are basically receptive followers but seldom leaders.

The next population chunk known as the late majority is also approximately 34% and are much more cynical about adopting these ideas and may do so out of economic necessity and/or peer pressure.

Laggards make up the last 16% of the population and have a strong belief in traditional values and how things "used to be done". These people are usually the most poorly educated, least worldly and lowest on the socio-economic ladder.

When comparing the diffusion of some innovations in other areas such as entertainment, business, military etc., educators, in general, would probably be classified as late majority and some may even be laggards.

This leads us to the crucial problem of trying to determine the necessary ingredients of innovation success. The most obvious problem is that there are so many situations and conditions that there will probably never be a certain success formula. The variances in professions and cultures mean that an innovation may be successful in one area/field and a failure elsewhere. Studies on innovation success sometimes have a difficult time pinpointing the true catalyst.

For example, the rapid development of the airplane may seem to have naturally occurred because of a transportation/time oriented public, but at the time, the Wright brothers' invention was not considered any competition for rail or ships. It was the Department of Defense who further developed this innovation right up through the jet engine and troop transport that truly paved the way for general aviation. Without the Defense interest at that time, aviation may have developed at a much slower pace and with a different result (Kopstein, 1971). Much of the success of computer based training can also be primarily attributed to Defense department support and secondarily to that of business and industry.

One of the most important ingredients of innovation success is that people must feel a need for a change. Rogers points out that the actual need for change is not even necessary but that attitude toward change is. "It matters little whether or not an innovation has a great degree of advantage over the idea it is replacing. What does matter is whether the individual perceived the relative advantage of the innovation."

This is one of the biggest problems in education. It has often been difficult to convince many educators that a problem exists or that improvement is necessary especially in times of tight budgets. In some cases, there <u>is</u> no problem or need for improvement. David J. Klaus (1969) said, "Not all innovations are particularly productive or worthwhile. One reason is that a number of these innovations have been created which, unfortunately represent solutions to non-existent problems...they failed to have an impact because they overcame no recognized difficulty."

Private industry also faces these problems. Quadrophonic stereo was introduced with the hope that as many users would shift to four channel sound as mono users did to stereo. The industry was surely frustrated as many stereo users could not hear that much difference to justify that much cost increase and, therefore, did not perceive the advantage or "need" to buy this innovation.

Another innovation died and later succeeded because the timing of this "need" dictated such. In the middle sixties, there was a light beer marketed called "Gablinger's" whose main campaign was a "less filling" type of approach.

It never succeeded and the product was quickly off the market. In the seventies, "Miller Lite" was introduced with a similar promotion emphasizing fewer calories. There are now several successful "lite" beers who emphasize low calories in a society that is much more calorie conscious than in the sixties. If the public perceives a "need", they will buy part of the "solution".

It is safe to say that much of the educational field does not perceive a need for computer-based instruction. When private industry faces a problem like this, they employ a change agent. A change agent must make people aware of the product and their need (or their perceived need) for it. A change agent could range from a U.S. Government registered lobbyist to hiring a local farmer to commit all their land to hybrid seed corn. Change agents can also come in the form of advertising.

Change agents are not as prevalent in education as they are in private industry. Education is not a very profitable market and innovators know it. Private industry may oblige the field with spinoffs as anything else would be too risky an investment.

One of the main differences between the educational field and that of private industry is that private enterprise invests comparatively larger amounts towards the research and development of change. David Klaus points out that education cannot afford this cost. He then directs us to a statement of H.F. McCusker and P.H. Sorrenson (1966), "Perhaps one of the strongest reasons for the slow change in educational practice is that very little is invested in conscious efforts to induce change. Education's research and development budget, relative to its total expenditures, is trivial."

Evaluating outcomes for both are entirely different. Private industry trys to induce a change to create a new market for profit which is an easily measurable goal. Educators must strive for a different goal of educational outcome which is much more difficult to measure and strive for (Goldberg, 1970).

Oettinger (1969) provides another example by relating the guidelines of Science Research Associates (SRA), a commercial organization that supplies educational materials. Before SRA will proceed with an innovation, they must determine an affirmative answer to three questions: 1. If it were available today, could they sell it to schools now? 2. Will it be developed in less than three years? 3. Will they recover their investment in less than five years?

Another important factor in the success of an innovation is how it affects or is accepted by the social system or culture. Introducing something like computer-based instruction into the strict Amish education system would be an example of innovation bound for failure.

Computer-based instruction is virtually a success in the military where it is adopted, heading that way in business and industry where it is basically in the Trial stage and just leaving the starting gate in education where it lies between Interest and Trial stages. As for most of its instructional technologies, education will utilize that which was commercially successful elsewhere. If this is true, then there will be a tremendous amount of education-oriented courseware to be developed.

When, if ever, will it be needed? Paul Mort's (1946) statement that "The average American school lags 25 years behind the best practice." may be the best prediction of when we can expect widespread education adoption of computer-based instruction.

References:

Anderson, C. (1962) Technology in American Education, 1650 - 1500. Washington, D. C.

Goldberg, Miriam, L. (1970) Evaluation of Innovations. Lowler, M.R. (Ed) Strategies for Planned Curricular Innovation. New York: Teacher's College Press, 51.

Klaus, David J. (1969) Instructional Innovation and Individualization. Pittsburg.

McCusker, H.F. and Sorrenson, P.H. (1966). The Economics of Education. Ross, P. H. and Biddle, B.J. (Eds.), The New Media and Education. Chicago: Adline Publishing Co., 185.

Mort, Paul R. (1946) Principles of School Administration. New York: McGraw-Hill.

Oettinger, Anthony G. (1969) Run, Computer, Run: The Mythology of Educational Innovation. Cambridge: Hardvard University Press.

Rogers, Everett M. (1962) Diffusion of Innovation. New York: The Free Press.

"GIGO"--How Does It Apply To Computer-Based Instruction?

Glenn E. Snelbecker, Ph. D., & Debra S. Stepansky, M. S.

Temple University

ABSTRACT

Diverse views have been expressed concerning instructional uses of computers. The present authors raise the possibility that "GIGO" -- "garbage in, garbage out" -- is applicable to instructional uses of computers, as has been acknowledged with regard to data analyses. Potential problems are identified. The potential contributions of authoring systems and languages are considered, but it is suggested that more appropriate help will come from contemporary approaches to curriculum and instruction development theories and models. Instead of dwelling excessively on computer technology, per se, the present authors contend that we must focus on educationally relevant, functional contributions if we are to have computer technology help to make instruction more effective, efficient and enticing!

CONTRASTING VIEWS ABOUT INSTRUCTIONAL USES OF COMPUTERS

Almost anything you read today concerning education or training makes some reference to the ways in which computer technology will be valuable. However, while many agree that the matter of instructional uses of computers should be addressed, there are divergent views about what should be said and done! Support for such ideas ranges from guarded optimism to unrestrained enthusiasm that many, if not all, of our problems in education and training will somehow be resolved if only we use computer technology. Critics of computer use range from those who express skepticism about realistic contributions to those who view virtually any machine-related innovation as somehow anti-humanistic.

In this paper we take a more moderate approach. While we anticipate that computer technology could make major contributions in education and training, we are concerned that such potential benefits may not be realized unless certain action is promptly taken. In brief, action is needed now to assure that computer technology will be employed in functionally relevant ways. Otherwise, computers may follow the course of various preceding instructional innovations (e.g., educational television, teaching machines, etc.) which were also promoted as having great potential contributions for education and training, but which did not fulfill advocates' claims.

PROBLEMS WITH COMPUTER USES?

What indications are there of problems concerning instructional uses of computers? Let's consider a few examples. One newspaper article had this headline: "Programmers get a failing grade on education software." According to educators, software specialists, and others quoted in the article, part of the problem lies in the minimal role which educators had in creating at least some of the courseware. This lack of input from professional educators, of course, may partly account for current problems, but there are indications that the matter is much more complex!

Other writers also have been expressing concern about instructional

uses of computers even when educators are involved in developing courseware or in planning ways to use computer technology. For example, George W. Bonham (1983) selected this title to express some of his concerns: "Computer mania: Academe's inadequate response to the implications of the new technology." Bonham suggested that, after some initial hesitation, educators are proceeding so rapidly that they are not carefully considering how computer technology might best be used: "First at a cautious trot and now at breakneck gallop, our schools and colleges are moving toward the promised land of computers. From all the present evidence, however, the longterm educational results are likely to be minimal and they could be catastrophic" (page 72). He contended that all too often there is a tendency to focus on the technology without considering adequately the long-term education objectives or the educational significance of what is being done with computer technology. Somewhat similarly, J. Raben (1983) cautioned that very soon (and even now, in some cases) we will be asked to explain and to defend the manner in which we are employing this potentially beneficial technology. Raben took the position (also endorsed by the present authors) that there is a distinct tendency to focus on what computers can do, rather than on what students and teachers need from an educational standpoint. Bonham offered an enlightening analogy: "It is as though, with the invention of food processors, we were determined to consume only that food capable of being sliced, diced, and pulverized. Such behavior would be of substantial benefit to manufacturers of food processors, but it would hardly improve the national diet" (Bonham, 1983, p. 72). With regard to instruction, there is too much of a tendency to focus on what computers can do best, rather than on ways in which the capabilities of computers can be employed to facilitate human learning.

Let's consider a few more critical comments before we explore some questions and suggestions that can help assure more appropriate uses of computers. In the April, 1984, issue of Psychology Today, Joseph Weizenbaum, a professor of computers and information science (CIS) at MIT, expressed concern that computers might have an undesirable effect on thought processes, as well as on aesthetic and moral judgments. Daniel McCracken, a highly regarded computer programming text author and CIS professor at CCNY, expressed similar concerns at the March, 1984, Association for Computing Machinery meeting in Philadelphia. He depicted many computer literacy courses as wasting time and possibly misleading students about what computers can do and what career opportunities exist with computers. In the March, 1983, issue of Today: The Videotex/Computer Magazine, Vicki Blum Cohen, a professor at Teachers College, Columbia University, depicted much courseware as mainly 'drill and drill,' with insufficient attempts to challenge students to think for themselves or to deal with concepts.

"GIGO"?

The title of this paper starts with the acronym, "GIGO." Those of you who have conducted statistical analyses of data may be aware of the terms and ideas involved: "Garbage in, garbage out." Whether we're talking about uses of elegant statistical models or uses of computerbased data analysis systems, GIGO signifies that we only "get out" good results if we "put in" sound data. We're suggesting that a comparable situation may exist with some, if not most, courseware and with other instructional uses of computer technology. In brief, it is the general thesis of this paper that we will fail to gain the potential benefits of computer technology unless such applications are sound with regard to curricular and instructional aspects, and are viewed as enticing both by students and educators.

HISTORICAL PERSPECTIVE

In planning how to deal with contemporary issues, it often is benefical to consider what was said and done in the past. A century ago, many made claims about potential contributions of the blackboard -- claims much like those being made today for computers. During the 1950s and the 1960s, advocates of teaching machines and programmed instructon identified various ways in which that technology could aid students and instructors. In each case, the promised benefits did not materialize as expected.

In 1959, W. E. Spaulding -- then, President of the Houghton Mifflin Company -- proposed that new teaching aids should be judged in terms of their contributions to improving the quality of education. We contend that his comments of twenty-five years ago are equally applicable to our contemporary, "1984" fascination with and concern about computers. A paper by Michael

Rossman (1984), a science teacher in an alternative school, illustrates why we need to consider carefully how we plan to use computers. After describing his extensive efforts to make certain that his school would be able to use computers, he found that he was losing interest in doing so, and that other teachers seem to have similar reservations. "Nor was I the only one dubious about the well-hyped advent of computers in schools" (p.13). Rossman went on to suggest that educators (and others) should ask "what educational values we should hold and support in relating to this new tool. To ask this question may lead us not only to resist many possible uses of the computer but also to a sharper sense of purpose and a more intelligent commitment to the uses we care to make" (p. 15).

PROPOSED QUESTIONS AND SUGGESTIONS

How can we avoid "GIGO" type problems with regard to instructional uses of computers? Here are some questions and suggestions relevant to uses of computers in education and training.

1. Authoring systems and languages?

To what extent will "authoring systems" or "authoring languages" help us to avoid "GIGO problems"? DATA TRAINING's 1984 Computer-Based Training Guide identifies 33 microcomputer-based systems. These 33 systems' features and capabilities seem to be representative of what we can expect from authoring systems or languages both currently and in the future. Judging from our own personal contact with some systems, reviews in the professional and mass media literature, and descriptions offered by vendors, it is apparent that these authoring systems as well as authoring languages, can be of some help. However, the help mainly consists of minimizing the need for computer programming skills and the need to attend to "housekeeping details." These systems, in contrast, do very little, at best, to address important curricular or instructional aspects. For example, with or without authoring systems, someone must decide what subject matter should be taught and in what sequence. Someone must consider what learner attibrutes are relevant for instruction and what practical steps are needed to deal with such attributes. Also, quite importantly, someone must be knowledgeable about the merits and limitations of alternative instructional strategies and techniques; someone must also plan how to use them so that the resulting learning experiences will be appropriate for the particular subject matter, learner attributes, and learning conditions.

2. Other places to seek help?

Where can one go for assistance on curricular and instructional matters? This is a question that was explored by the senior author on several previous occasions (e.g., Snelbecker, 1974; Snelbecker, 1983). Throughout most of the past century, there has been interest in delineating ways whereby research findings and theory could be used to improve the quality of instruction and learning.

However, it is only within the past few decades that systematic approaches to curriculum and instructional development have found general support. In part, these efforts were supported by federal government programs starting in the 1960s. But also of significance were certain technological advances. More specifically, the "teaching machine / programmed instruction" technology of the 1960s, and the mainframebased CAI ventures starting especially around the late 1960s and early 1970s stimulated interest in devising more systematic means for producing and evaluating curriculum and instruction. Currently, the availability of quite powerful microcomputer systems at affordable prices has renewed general interest in theory and techniques relevant to curriculum and instruction design and development. As a result, there are emerging today much more comprehensive approaches to instructional design and development than existed even a few decades ago (cf. Reigeluth, 1983; Snelbecker, 1974).

Despite the fact that many instructional systems design ("ISD") ventures mainly (or, merely?) emphasize measurable learning outcomes, contemporary instructional design models offer guidelines for recognizing and dealing with behavioral, cognitive and affective processes. These models provide systematic means for checking consistency across curricular goals, instructional events, and evaluation procedures. Thus, they provide systematic means for designing instruction that is both internally coherent and compatible with pertinent learner attributes and learning conditions. Moreover, much more than was the case even a decade ago, these instructional design theories and models are based on a growing body of research literature. One major effect

is that educators and students now theoretically could have access to a range of effective, efficient and enticing learning opportunities. However, whether they actually will have access depends on the extent to which producers use currently available instructional design principles and the degree to which potential users consider such principles in selecting learning resources.

3. Relationship with selection of other learning resources?

Must one be a "computer expert" to design instruction? The present authors contend that computer expertise, per se, is not only not mandatory but also that knowledge about computers will not necessarily be adequate as a basis for planning good learning experiences! (For vivid examples, check almost any "documentation" for software or hardware!) Such a view typically leads to another important question: Are the principles, theories and models that can be used to plan computerbased instruction applicable to learning resources that do not involve "computer technology"?

They are relevant for education and training in a variety of contexts, with learners of different ages and abilities, and with diverse kinds of subject matter. During the past few years, the present authors have been reviewing literature concerning emergent instructional design theories, research on teacher effectiveness, and views about means for improving the quality of instruction. For example, we found that there are many similarities between attributes found with effective classroom teachers and key principles embedded in contemporary instructional design theories and models (Snelbecker & Stepansky, 1983). Thus, to avoid "GIGO problems" when making plans for instructional uses of computers, one can use the same kinds of principles that are applicable in sound selection of learning resources more generally. The senior author has aided hundreds of undergraduate and graduate students, as well as industry trainers, to use their understanding of education and training to make informed choices about functionally relevant uses of computer technology.

CONCLUDING COMMENTS

It is the main thesis of this paper that we can avoid "GIGO problems": We can expect to enjoy the considerable benefits of using computers in education and training if we take certain steps. We must consider the educational relevance any time we consider using computers. This can be done by learning about and using sound instructional principles. But use of these instructional principles requires that we synthesize and integrate information from diverse sources. We must overcome the fragmentation that occurs when rather narrowly defined topics are addressed in research and theory. We must integrate with these instructional principles an understanding of the inherent nature of the subject matter to be learned and should identify relevant attributes of the learner. We must also overcome the fears and the mysticism which frequently emerge when people talk about computer technology. We must focus on the educationally relevant, functional contributions which computers may offer to make education and training experiences more effective, efficient, and enticing!

REFERENCES

Bonham, G. W. Computer mania: Academe's inadequate response to the implications of the new technology. The Chronicle of Higher Education, March 30, 1983, p. 72.

Gerber, C. H. Educational computing: There's a lot to learn. Today: The Videotex/Computer Magazine, March, 1983.

Kemske, F. (Ed.). Data Training's 1984 Computer-Based Training Guide, Boston: Warren/Weingarten, 1984.

McCracken, D. D. A skeptical view of computer literacy. Paper presented at the meeting of the Association of Computing Machinery, Philadelphia, February 1984.

Raben, J. Advent of the Post Guttenberg University. Academe -- Bulletin of the AAUP, 1983, 69(2), 21-27.

Reigeluth, C. M. (Ed.) Instructional design theories and models: an overview of their current status. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1983.

Rossman, M. How to use the computer in science class (and how not to), Classroom Computer Learning, February, 1984, 13-18.

Snelbecker, G. E. Learning theory, instructional theory and psychoeducational design. New York: McGraw-Hill, 1974.

Snelbecker, G. F. Is instructional theory alive and well? In C. M. Reigeluth (Ed.), Instructional design theories and models: an overview of their current status. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1983.

Snelbecker, G. F., & Stepansky, D. S. A review of the instructional design literature as it relates to classroom teaching. Commissioned Monograph submitted to the National Teachers Examinations Office, Educational Testing Service, Princeton, New Jersey, 1983.

Screen Design for CAI

Timothy W. Spannaus, Ph.D., Creative Universal, Inc.

Collette Pariseau, Goal Systems, International

ABSTRACT

To the learner, a CBT lesson really consists of the screens that he/she sees. As designers, we may not pay adequate attention to the design of screens, preferring to concentrate on the design of the whole lesson or even the course. It is essential that each screen be attractive, easy to understand, and instructionally effective. This paper provides general guidelines for the organization and layout of CBT screens based on educational psychology and the experience of typographers.

Screen Design for CAI

Introduction

Many of us in the instructional design business pay scant attention to the actual appearance of instructional materials. That is not to say that we don't recognize the importance of appearance, but in many organizations, we research and write the instructional product, and leave the page layout and design to artists, illustrators, art directors, and others with professional preparation and experience in art and design. We may not have that luxury in development of CBT programs, because of differences in the way such programs are put through the shop.

In development of traditional instruction, developers concern themselves with the organization and content. The manuscript is passed to the production department to be put into final form. The instruction[illegible]ner reviews the work, but isn't [illegible]volved in its production. [illegible]opment of CBT materials, [illegible] projects do not go to produc[illegible]. These projects lend themselves well to on-line development or input by clerical personnel, so that the developer has complete control over the appearance of the final product. We should be able to assume that the developer has some understanding of the instructional principles relating to screen design, but we cannot assume any esthetic sensitivity.

We will consider both instructional effectiveness and esthetic judgments to be important. Fortunately, there is rarely any conflict between them. But, obviously, if a conflict develops, the instructional considerations have to take priority.

Among important instructional considerations are the predictable ways in which the learner's eyes move across the screen. In general (at least among those using the Roman alphabet), people's eyes move in an arc from upper left to lower right (see figure 1). When actually reading text, of course, a detailed trace of eye movement would reveal a series of Z-shape movements, as the eyes follow lines and retrace back to the left margin. But the overall arc movement cannot be ignored. Advertisers use the lower right corner of magazine or newspaper ads for the company name, logo, or a reply coupon, because the reader's eye lands there after scanning the ad.

As a consequence, we should use the upper corners and the lower right corner of the CBT screen for information the learner must see, since the eyes will usually scan those areas. Do not use the lower left corner for information that must be noticed. You can use the lower left corner for information the learner has reason to look for, such as prompts.

But do not use it for warnings or messages that appear only occasionally.

Perception

Before learning can occur, the learner must perceive the text or graphics on the CRT screen. While each of us perceives things in unique ways, research has produced some useful generalizations.

The eye is attracted by contrast. A portion of text or graphic that is different from the other text or graphic on a screen will draw the learner's eye. The difference may result from use of a contrasting color, larger type or a different character set, highlighting, or reverse video-whatever the CRT and software allow. However, the effect of contrast is lost if it is overused. Highlighting of a third or more of the text becomes an annoyance, rather than an attention-grabber. At most, one or two key words or phrases should be emphasized.

If you use color for emphasis, use colors that contrast well for all users. In one case, we used a green graphic for emphasis with the remaining graphic in yellow. Our color-blind client wasn't aware of the color difference. Avoid using red and blue together if the learner's eyes will have to focus on them both. Because these two colors are distant from each other in the spectrum, the eye cannot focus on them simultaneously.

Memory

If learning really takes place, newly-learned skills and information will reside in long-term memory, and the learner will be able to retrieve them efficiently.

Learning theorists generally agree that information is more easily recalled when it is efficiently organized in memory. For example, learners seem to be able to separate general statements and examples of the generalizations, and deal differently with them. Most learners retain and can easily recall the essentials of the generalization, but few of the examples. In CBT, the formation of the cognitive structure is significantly aided by the layout of each screen.

If we expect the learner to readily separate information by its function, let's help by using the screen layout, labeling, or other techniques to indicate the separation. One such technique is varying the left margin. Adult learners have long been conditioned to interpret left margins in predictable ways. The use of the margin indicates a hierachy -- information further to the left is perceived as being more important or more general, as in an outline. So a paragraph containing a generalization might begin at column 5, but a paragraph containing an example might shift over to column 10 (see Figure 2).

Learners oriented to an outline structure also perceive information higher on the screen as being more important than information lower on the screen.

(Note: This discussion assumes that left margins are justified or lined up evenly. Left margins should always be justified, so eye movement is uniform and predictable. But right margins should not be justified on CRT screens. In general, ragged right type is easier to read that right-justified. There are some exceptions to this generality, but they do not apply to the vast majority of terminals used for CBT).

Another useful structuring technique is the use of "white space." That term is one used by printers and typographers to describe the blank areas around text and graphics. On a CRT, we are more apt to have black space or even green space, but "white space" is still a usable term because it is readily understood.

The white space is used to set off various portions of text. Even when paragraphs are indented, there should be a blank line between them. Information that requires special emphasis, or which interrupts the flow of the text (like an example) may be set off by extra blank lines.

Overlays

The division between perception and memory is a little artificial, since both are critical and some screen design problems affect both. You probably know that you shouldn't fill a screen with information. The learner needs the white space to organize his or her perception and to structure the information.

But what do you do when you have a lot of information that is closely related? Screen "overlays" (or "rewrites" or "over-writes") allow you to leave some information at a screen while changing the rest. For example, you may need to explain several different aspects of a complex graphic, or provide many directions for use of a simulation.

Overlays can be confusing, however. To make sure the learner knows that a portion of the screen is new, you might alter the shape of the text or whatever you overwrite, so even a casual glance will reveal the change. A timed pause between the erasure of the old material and writing the new also make the change obvious.

For an overlay to be most effective, the new and old content should be parallel in function. That is, you should overwrite an example with an example, or directions with more directions. The learner will then be able to use the content in a predictable, intuitive way.

Facilitators

When you have determined the layout of the instructional content of the screen, you can add facilitators, or aids, to each screen to assist learning. These facilitators are derived from typography-or the graphic art of placement of print on a page.

In developing CBT lessons, we must divide our lessons into fairly small blocks, to fit the screen. Also, most CBT systems do not provide easy means by which to page back and fourth through a lesson. These constraints require that each screen be complete on its own.

One facilitator, the footer, tells learners where they are in the course. It can be placed in the lower left corner of the screen (Figure 3, Location 4). Because this position is the last place learners look on the screen, the footer will not be distracting to the learner, but can be used for reference when needed. An example might be "Tutorial 1: Check Acceptance."

Another aid, the screen label, tells the learner the instructional intent of the screen. Labels are especially critical in CBT since many screens are restart points. That is, learners may leave a lesson before the end and return to the same topic or screen from which they signed off. Labels will help them to re-establish the content of the lesson. Because of the importance of labels, they should be placed in the upper left corner of the screen so students will see them before they continue the lesson (Figure 3, Location 1). In the same lesson for which the previous footer was written, the screen label might be "Authorized Signature."

Without an instructor to answer students' questions, CBT must include many directions. Directions should be set off from other material on screens so learners will not confuse the instructional content of the less with directions on how to proceed through it or respond to a test item. Drawing boxes around directions will set them apart (Figure 4). Depending on the character set available on the system you are using, you may draw the box with a specified charater such as an asterisk, or just with lines.

Student aids are only helpful when students know how to use them. Make sure you tell learners that you are using footers to give them a point of reference in the course; labels to specify the instructional intent of the screen, and a specific format, such as a box, to specify directions. Always use the same format, too. Varying the placement of footers and labels, or the method of setting off directions, will only confuse student.

Careful attention to the memory and perceptual requirements of your learners will help you design CBT screens that are once aesthetically pleasing and instructionally effective. Screen designs based on the principles of perception and memory will not only be pleasant to look at they will help learners to organize, remember, and recall the instruction they receive.

References

Fleming, M. and Levie, W.H. Instructional Message Design. Englewood Cliffs, Educational Technology Publications, 1978.

McConkie, G.W. "Learning from Text" in Review of Research in Education No. 5 (L.S. Shulman, Ed.), Itasca Illinois, F.E. Peacock Publishers, 1977.

Merrill, P.F. "Displaying Text on Microcomputers," in Johassen, D.H. The Technology of Text, Englewood Cliffs, Educational Technology Publications, 1982.

Reynolds, L. "Display Problems for Teletext," in Johassen, D.H. The Technology of Text, Englewood Cliffs, Educational Technology Publications, 1982.

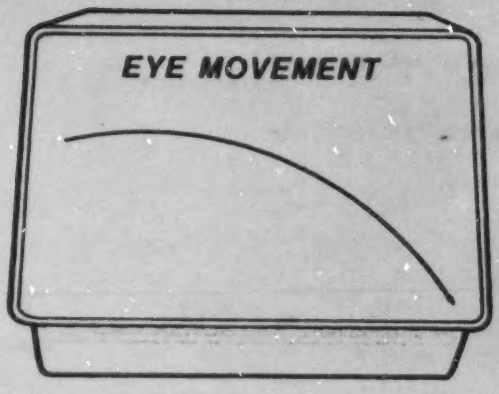


Figure 1. Eye Movement

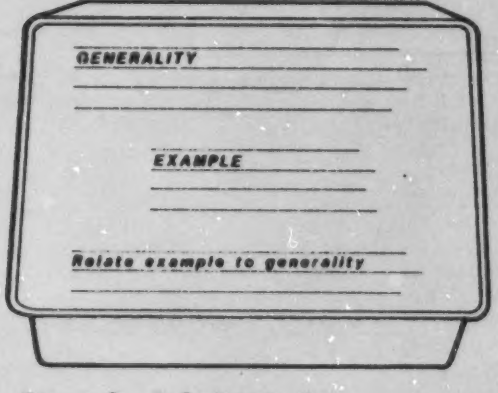


Figure 2. Left Margin Use

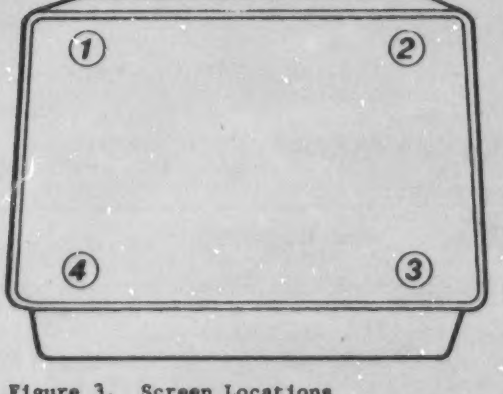


Figure 3. Screen Locations

COMPUTER-BASED INSTRUCTION AT THE U.S. ARMY ENGINEER SCHOOL

EVOLUTION AND MANAGEMENT OF AN IN-HOUSE CODING TEAM

Captain Charles L. Stone
United States Army Engineer School

ABSTRACT

In late 1983 the U. S. Army Engineer School began an in-house project to develop PLATO courseware for the Engineer Officer Advanced Course. In the year that followed, much was learned about the organization, resources, and process needed to produce courseware. The temporary nature of our workforce has presented stiff challenges to our lesson development efforts. In particular, the training and supervision of coders (programmers) has required creative approaches to personnel recruitment, training, and supervision. This paper documents the ongoing evolution of lesson development, and in particular, the coding process at the Engineer School. The solutions we have discovered may be valuable to other organizations proceeding with in-house development of CAI.

BACKGROUND

In early 1983 at Fort Belvoir, Virginia, a project was begun to restructure the curriculum of the Engineer Officer Advanced Course. A temporary team of Army Engineer officers was assembled to design and implement the new course. In addition to reviewing the curriculum of the Advanced Course, the team sought to include the use of mentors and computer based instruction. Early research into CBI resulted in the decision to adopt the PLATO system and to proceed with in-house courseware development. Little was known at the time about the level of resource investment required to support in-house lesson development. A goal of "100 hours of CAI" was established for implementation with the new course on 1 October 1984.

During Oct 83 one permanent staff member was assigned to supervise the CBI effort. Four Army Captains and one civilian Educational Specialist were detailed to him for a period of up to six months to identify, design, and code lessons. No one had any prior experience developing CAI. A review of current literature was done to construct a model for the development process and to compile a knowledge base of CAI design principles. Design of the first lesson began in December 1983. The coding of that lesson began the following month.

DEFINITIONS

Throughout this paper, 'Computer-Based Instruction' (CBI) is used to describe the integrated combination of 'Computer-Assisted Instruction' (CAI) and 'Computer-Managed Instruction' (CMI).

Engineer Officer Advanced Course (EOAC): A six month resident course of instruction for Army Engineer Captains. The curriculum contains instruction on many military engineering subjects such as bridge, road, and airfield construction; as well as general military subjects.

Courseware Development Cell (CoDeC): The group responsible for the development and implementation of CBI in EOAC. Currently, CoDeC consists of a design team and a coding team.

Designer: An individual in the design team who coordinates with the SME, develops lesson strategy and flow, and creates storyboards.

Coder: An individual in the coding team who converts storyboards into TUTOR code so as to create a working CAI lesson.

Subject Matter Expert (SME): The instructor in the Engineer School who teaches the subject and acts as the expert for the Army in his particular subject area.

EVOLUTION AND MANAGEMENT OF AN IN-HOUSE CODING TEAM

EVOLUTION OF THE COURSEWARE DEVELOPMENT CELL

At the end of 1983 the courseware development cell was organized into three sections. A design section of two individuals coordinated with the SME's, developed the lesson strategy, and wrote the text with instructions for graphics, branching, and judging. A storyboard section of three individuals took the script developed by the designers and created storyboards. A coder section of two individuals coded the storyboards into the TUTOR language. Under this system the storyboarders soon realized that they were simply copying the text provided by the designers onto storyboards and adding graphics. We were wasting time and talent.

In late January 1984, the group reorganized into two teams. Each team was responsible for the design, storyboarding, and coding of a lesson. The goal was to enrich the jobs of each member by having all team members share in each development step. Member boredom and duplication of effort was reduced. The significant weakness in this team approach was that each team member would have to be proficient with the TUTOR language as well as lesson development strategies and design principles. We could not expect everyone to be an expert in all phases of lesson development.

In February, 1984, we again reorganized. Two sections were created; a design team and a coding team, each of three to four individuals. Group members were assigned to a team based on their interest and demonstrated talent. This organization has worked well for the past eight months. Duplication of effort is almost nonexistent, members are generally satisfied with the scope of their responsibility, and yet there is enough specialization for efficient operations.

THE CODING TEAM

The temporary nature of the coding workforce has made our group different from others in the CAI field. Since late February of 1984 the coding team has ranged from three Army officers to five full-time and three part-time military and civilian personnel. Except for the team leader (the author), members have been assigned for only six months. Five coders have already left the group and two more are scheduled to leave this fall. The supply of temporary officers has been terminated by higher headquarters. Faced with a shrinking workforce, we were forced to look elsewhere for new members.

A solution has been Veterans' Administration training programs for disabled veterans. Currently we have one veteran working full time. He is paid by the VA and is expected to work with the team for three to four months. At the conclusion of his training he may be hired as a permanent employee.

Another source of employees is the Catholic University which has a graduate program in 'Curriculum Design and Educational Technology.' Through a cooperative program with CU, we have arranged to hire students to work 20 hours per week.

NEW CODER TRAINING

The temporary nature of our workforce has presented a stiff challenge. The goal is to teach new coders rapidly so as to get the maximum production before they leave. Four approaches have been tried with varying success:

a. Introduce new coders to PLATO and give them a week or so to get acquainted with the features of the system. Show them some lessons which demonstrate capabilities of PLATO and then give them some storybards to begin coding. An experienced member is assigned to answer questions, give suggestions, and critique their work. Initially, they are given simple storyboards. As they learn, animation, branching, and judging are added. This approach generally works only for those with prior programming experience.

b. Another approach to new member training is to use a highly structured learning vehicle. PLATO offers the CREATE curriculum which includes CAI, textbooks, and practical exercises. We have chosen to introduce our VA employees to the TUTOR language via parts 6 and 7 of the CREATE curriculum which teaches the PLATO Author Language (TUTOR). A real advantage of this approach is that once started, new coders can pace themselves without much supervision or assistance from other members. A disadvantage is that about four weeks is needed before the individual has completed the course and is ready to begin coding lessons.

c. The approach most often used is a combination of the above methods. New members are started in the CREATE curriculum, Parts 6 and 7 on the PLATO Authoring Language, and spend about two weeks learning the basics of the TUTOR language. The use of reference manuals and AIDS is encouraged. With a cursory knowledge of the language, individuals can start coding portions of a lesson and, when questions arise, know enough to understand the

specifications and the examples in on-line or off-line reference manuals.

d. Possibly the most desirable training strategy is to send new coders to an intensive TUTOR training course of one to two weeks. A one week course was provided to the first two CoDeC members and I would recommend that it be given consideration. Following the training course, our coders had enough knowledge to begin coding and seek assistance as needed.

THE CODING PROCESS

The early coding of lessons was done using ID/SD to create graphics and then adding nondisplay instructions and documentation. The real disadvantage of this approach is the ID/SD editor which is painfully slow and awkward, although it does have some practical uses such as scaling and rotating displays.

In December a Control Data Corp. consultant introduced us to PCD2. PCD2 is a menu driven authoring tool which does not require the user to know the TUTOR language. We did considerable experimentation with PCD2 and coded portions of some lessons. We did decide, however, that for our current applications, PCD2 was not suitable. The major drawbacks were the limited branching keys available to students and the difficulty of judging numerical responses with units, for example: 120.4 lbs/ft^2. Feedback for multiple anticipated wrong answers was also awkward. In the future, however, PCD2 may be used in the Engineer School by instructors to create less complex CAI materials.

A strong feature of PCD2 is its graphics editor which turns out to be based on the GUIDE editor. GUIDE is a system utility for the creation of screen displays and is generally superior to ID/SD. Since January our general coding procedure has been to create several storyboard displays in GUIDE, convert them to TUTOR code, and add judging, branching, documentation, and other features. We are pleased with the success and efficiency of this system. Occasionally ID/SD is used to scale or rotate some part of a display and to create touch areas.

We have two graphics tablets which we use with the system utility =0gis=. The tablets are less useful than expected because of the time needed to setup =0gis= and load and unload the terminal. Also, the graphics done on the bitpad tend to get too detailed and take up a tremendous amount of file space. We use the bitpads only for complex drawings. Most graphics are created in GUIDE.

Currently, the coding process begins when the senior designer passes a completed design to the coding team. The lesson is reviewed by an experienced coder before any coding begins. This review is done to identify problems in the lesson which would confuse the coders, or cause discrepancies or inconsistencies in the lesson. Often, branching or judging instructions are incomplete or unclear. Another common problem of inexperienced designers is the attempt to put too much information or impractical illustrations and animation on the screen. Written comments are passed back to the senior designer for consideration, clarification, and correction.

Once the intent of the designer is clear, coding can begin. A key step in efficiently managing the coding of a lesson is the setup of the initial parameters and standards. For each lesson coded, the first step is to create a 'startup' block which loads a character set; sets up, defines, and initializes variables; and enables or disables system features which will be available throughout the lesson.

Standard coding practices have been developed to aid the efficiency and consistency of lesson coding. While it is not the intention of this paper to teach the fundementals of sound programming, one should recognize that much attention needs to be given to the development of coding standards. The sample code below and on the following page represent some standards in use at the Engineer School.

Typical 'Startup' Block

```
lvars     10
common  $$ declare and load any common
storage $$ or storage to be used
*
imain     setup    $$ declares imain unit
finish    finisher $$ declares finish unit
*
define  $$ sets up global define set
        $$ only define command per lesson
        $$ all variables documented
        $$ no primatives used in lesson
*
setperm $$ initialize/randomize variables
*
do        cloader('lesson','block')$$loads
*          character set & displays message
*box/draw to display screen border, if any
*
unit      first  $$ avoids -jump- in ieu
*
jump      unitl $$ to first storyboard
use       lesson,block  $$utility lesson
```

Subroutines help standardize the coding of a lesson. When an illustration or paragraph or judging routine is used repeatedly throughout a lesson, a subroutine is created. Not only is this an efficient coding practice, but it assures uniformity throughout the lesson. One member may be assigned to produce all of the graphics for a lesson.

When more than one coder is assigned to the lesson, it is best to assign whole sections of a lesson to a coder rather than one or two storyboards at a time. The reason is that if storyboards are assigned piecemeal, then the individual coders will not be able to run sequences of displays to check for correct transition from display to display.

Standard Unit Structure

```
unit      name $$ same name as storyboard
          ctr  $$ local variables (docum.)
*
*  Description of unit (from GUIDE)
*
restart   $$ if any
area      $$ if any
*
back   \
data    >$$  any branching keys enabled
help   /
*
do        eraser $$ erases inside border
*
*body of code
*
jump      nextunit $$ always jump (or end)
```

New coders are assigned to work with an experienced member. The new coder creates displays in GUIDE. His coach checks his work and adds the branching and judging instructions.

THE DESIGNER-CODER RELATIONSHIP

During the joint designer-coder review of the completed lesson, as well as during coding, questions arise about the design of the lesson. Usually, the coder can check with the designer and get the question answered. Often disagreement arises about the design and the senior coder will settle the issue with the senior designer. The senior designer has the final authority on lesson design.

THE FUTURE

Ahead of us lie many challenges. No lessons have been validated in student trials. Many other lessons are coded, but must go through a design review before validation can begin. Still more lessons are in various stages of design.

All of the experienced designers and coders (except the two team leaders) will be leaving in the next two to three months. Replacements will be part-time graduate students from the Catholic University and veterans. The number of authorized full-time personnel will fall on 1 October to four individuals to handle design, coding, evaluation, CMI, and implementation.

Screen Design Guidelines

William G. Sweeters, Ed.D.

Ford Aerospace & Communications Corporation

ABSTRACT

Every CAI author is involved with the issues of screen design. This paper presents specific guidelines concerning the use of space, text, graphics, color, lesson control, and lesson orientation. These guidelines can be used as a starting point for instructional designers wishing to develop their own screen design standards.

The guidelines herein are presented positively in order to impress the designer with their function as a decision-making tool. They represent general suggestions for dealing with various content areas, students, and environments. I suspect that each of these factors may require special guidelines which will be learned through experience. A designer may choose to violate a guideline but should have a reason for doing so.

The best type of experience is to develop CAI lessons which are formally validated. Another good source of experience stems from the advertising industry. Designers can acquire a feel for aesthetically pleasing layouts which effectively convey information by paying special attention to contemporary advertising and poster techniques.

The Use of Space

Guidelines:

- Use space freely.
  A major difference between a CAI display and a text page is in the cost of space. While publishers try to introduce as much material as possible on a page, this restriction does not apply to CAI because space is practically free.
- Establish functional areas on the screen.
  Divide the screen into specific functional areas and use them consistently. Here are types of presentations which could be assigned to specific screen areas -
  Orientation
  Text
  Menus
  Graphics
  Student Directions & Responses
  Error Message
  Feedback & Prompts
  Student Options Menu

It is not necessary to rigidly adhere to specific dimensions for functional areas. Allow them to expand or shrink according to the amount of information that must be displayed while keeping their relative positions intact.

- Arrange text and graphics into blocks surrounded by ample spacing
- Consistently use and label areas on the screen to aid in cognitive structuring (i.e., if you are presenting a list of things, display them as a list - graphics
  - audio
  - video
  - video-disc

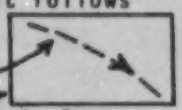
If you are presenting an outline, use indentations -

1. Most important
   a. Less important
      i. Least important)

- Consider the effects of timing to enhance learning.
  A quality of CAI is the ability to add and remove information at different times. Certainly it is difficult to concentrate on one

part of the screen while information is being added elsewhere. Therefore, do not add information until you want the learner to look directly at it. You might consider displaying an entire screen at once to avoid timing problems or you might give the learner control of the rate of display.

- Avoid the scrolling of text.
- Present blocks in this order -
  1. Graphics - so they don't distract from text
  2. Text
  3. Directions
- Place information so it follows normal eye movement:



information here is less easily perceived. Use this space for less important information or as a functional area.

- Erase blocks in a manner which guides the student's eye.
  Consider using WIPE - clears a screen slowly in a definite direction
  WIPE with FILL - fills in a pattern behind the wipe
  POP - erases instantly

Text

- Each display should fully use CAI capabilities by including at least one of the following:
  a graphic
  highlighting or symbols
  blank space to enhance text
- Develop symbols to emphasize standard prompts (i.e., ▶ means make a response).
  - Keep symbols simple and unobtrusive
  - Use them consistently.
  - Leave a blank space around symbols so that they can be animated in any direction.
- Present all instructions before presenting a cursor or arrow which requires a student response since students stop reading when they encounter a cursor or arrow which requires a response.
- Generally use lower case text with decenders. This is the easiest kind of text to read.

Justification

- Generally center the longest line in an area and left justify the others.
- Place text where it's most effective, do not always left justify it on the screen.
- Do not double (right and left) justify text.

Highlighting

- Limit highlighted items to no more than three (3) per display.
- Use blinking and flashing very sparingly. These devices totally capture attention and quickly become disconcerting.
- To emphasize text, consider the following:
  boxing
  underlining
  italization
  reverse video
  boldface
  capitalization
  multiple attributes
  e.g., capitalization with reverse-video

Word Usage

- Use short lines with a maximum of eight (8) words.
- Keep natural phrases on a line.

Not this:

Make sure the large drag
rake locks when the ladder is down.

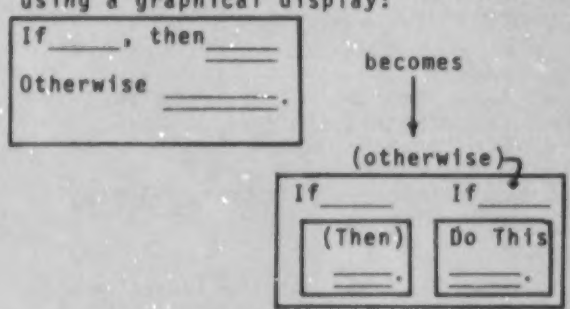
But this:

Make sure the large
drag brake locks
when the ladder is down.

- Consider using overlays to add a bit of information at a time.
- Never cover a page with text.

Graphics

- Consider using a graphic on each display.
- Whenever a message contains a conditional statement (e.g., if, when, otherwise, therefore, .), consider using a graphical display:



- Consider employing all of the following types of graphics:
  Realistic - portrays an instance or example of a concept. A realistic graphic.
  Analogic - relates a concept to more familiar, similar things. A more far-fetched but personal graphic.
  Logical - a highly schematized visual such as a flowchart, graph, or map. An abstract but logical representation

Most existing CAI lessons do not use enough graphics. When graphics are used they are usually the realistic type. Designers need to consider using analogic and logical graphics as well. (See Figure 1)

Three Types of Graphics to Support the Concept of "Tuning a Signal".

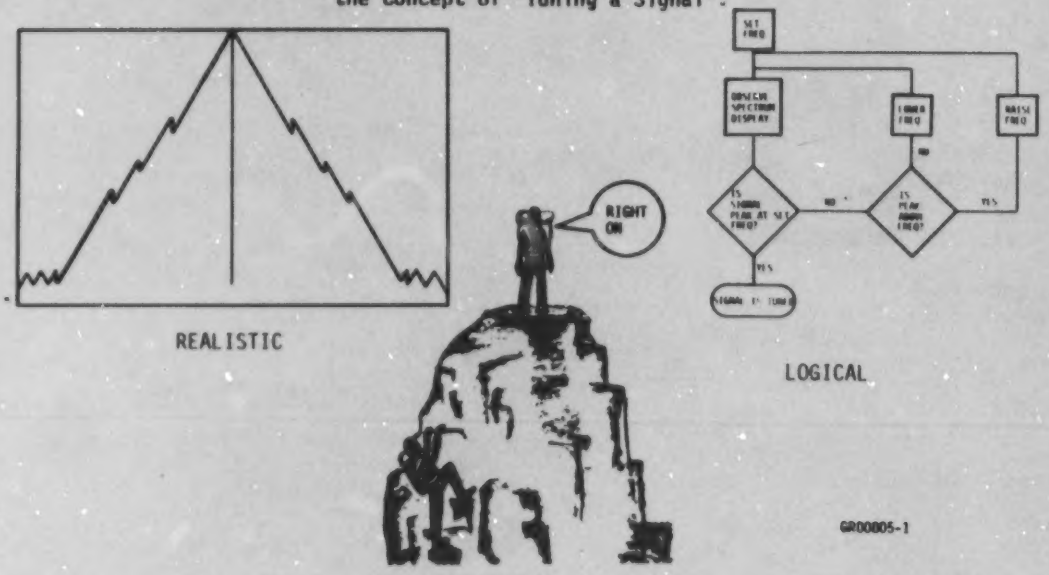


Figure 1

Use of Color

- Limit the number of major colors used in a lesson to no more than seven.
- Avoid bright colors, especially in backgrounds.
- For scenes, consider using a pallette of harmonious subdued colors.
- Do not use hot, vibrant colors for text display. These are colors that appear to "jump" when viewed.
- Choose a subdued color (neutral gray) and use it consistently as a background for text.
- Do not use more than four (4) colors in a display.

Lesson Orientation

- Try to approach the kind of orientation a student normally has with a textbook by including information which answers these questions:
  - Where am I in this lesson?
    How much more must I do?
    Where can I go from here?
    How can I get there?
  - provide course, unit and lesson maps.
  - On each frame, indicate the course, unit , lesson, frame and number of frames left to complete.
  - Provide menu options for student to get help, review, and exit the lesson.

Lesson Control

- Try to approach the kind of control a student normally has with a textbook by including the following options:
  - provide menus under student control.
  - allow students to control lesson presentations through the use of keys for HELP, BACKUP, GLOSSARY, REVIEW, and COMMENT.
  - allow students to control the rate of presentation. (i.e., Use PRESS [→] to continue.)
  - include branching to alternate lessons.

Alesandrini, K. L., "Instructional Graphics in Commercial Courseware." Proceedings of the 25th International ADCIS Conference, May 1984, pp. 188-195.

Digital Equipment Corporation., Introduction to Computer-Based Education. Author, February 1984.

Haines, J.M., Screen Design Strategies for Computer-Assisted Instruction. Bedford, Mass.: Digital Press. 1984

Hoekema, J., "Interactive Videodisc: A New Architecture." Performance and Instruction Journal, November 1983, pp. 6-9.

McDonnell Douglas Corporation, Author, AIS Style Manual. McDonnell Douglas Corporation, 1983.

Poriseau, C., and Spannaus, T., "Aesthetics and Instruction, Principles of CBT Screen Design." Data Training. March 1984, pp.41.

Steinberg, E.R., Teaching Computers to Teach. Hillsdale, N.J.: Lawrence Erlbaun Associates, 1984.

Orthographic Projection and Isometric Drawing Tutorial

Using Computer Graphics

Manjula B. Waldron, Ph.D., Mark R. Hopkins,* Ph.D.
Department of Engineering Graphics
The Ohio State University
Columbus, Ohio 43210

ABSTRACT

In introductory engineering graphics courses the instructor is required to teach concepts of orthographic projection and isometric drawing. This is done through a large number of drill and practice exercises using paper and pencil The instructor must grade these post haste so that the incorrect concepts are not internalized. Thus placing a tremendous grading burden on the instructor which has limited impact on student learning. To alleviate this problem and to provide drill and practice, special software was developed. Through this software the student was provided with a rectangular or isometric grid on the screen and a rubber band cursor controlled by a joystick. The entire program was menu driven and easy help files were available to the student. The students were assigned problems and their reactions were evaluated. The use and applications of this software is presented in this paper.

INTRODUCTION

Engineering drawings contain isometric and orthographic projection representation of objects. Engineering undergraduates in most academic programs are required to learn these skills on an ad hoc basis or as is most often the case yet, in an engineering or design graphics course. Representation of objects in the three principal views requires a visual abstraction that many students have a difficulty in grasping. Recent introduction of computer-graphics equipment by way of CAD/CAM in most industry has made it imperative that graduates from engineering curriculum be "visually literate". They need to not only conceptualize their design but also be able to break it down into three principal views. They also need to be able to communicate to the manufacturing personnel. The usual skill expected to be learned by the student in the classroom is to be able to take the isometric drawing of a three dimensional object and break it up into its three principal orthographic views of TOP, FRONT and PROFILE views. The inverse skill of taking the three principal views and constructing the three dimensional isometric drawing is also expected to be mastered.

Traditional classroom technique involves for the instructor to give a lecture demonstration and give paper and pencil practice exercises to take home (Earle, 1983). This works well for the students who understand the concepts quickly, but most of the novice students have trouble in performing this visual gymnastics initially. As a result they take a lot of time to complete the drawing before they submit it for grading. Two days later (by this time they have forgotten most of what they used) they get the corrected copy back. The instructor often doesn't have time to go over the student errors individually in detail. The result is frustration on the part of the students who are having difficulty and the instructor who has to grade and return the work without delay and with limited explanation.

To alleviate this problem a user friendly, easy to use, menu driven software was developed to assist students who were having difficulty. Practice exercises were put on the system so the instructor could assign these to individual students or class as a whole. This software and its classroom evaluation will be presented in this paper.

HARDWARE AND SOFTWARE

The system consists of IBM 3277 GA terminals attached to the tektronix 618 storage tube for graphical output. The terminals support joystick, light pen and digitizer tablet. This workstation has relatively low cost and high resolution for engineering graphics type of work (the line thickness can be varied and one doesn't need to sacrifice the line quality obtainable from a pencil). Each workstation was connected to IBM 4341 running VM/CMS and RPQ graphics package. Each workstation was also connected to a tektronix hard copy printer.

The software developed consists of the driving software and a three dimensional data base. The main program displays a user selected menu. The menu is accessed through the digitizer tablet and once the appropriate selection is made the student can then proceed to make his/her drawing. The student does not need to know VM/CMS system commands except to logon and logoff. The profile exec file along with the program controls the entire menu' structure. All prompts appear on the terminal. A help command on the menu to explain each step is available to the student at all times. When the student logs on, the profile exec file (through a menu) allows the student to call up the OP or the ISO DRAW software. The student makes the selection of either drawing the orthographic projection given the isometric drawing (OP DRAW) or drawing the isometric drawing given the three orthographic views (ISO DRAW). The problems menu consisting of 20 problems is then displayed. The student selects the one he/she wants by using a light pen. The software accesses the appropriate data files and displays the problem statement.

Let us assume the student selected OP DRAW and the assigned problem. The isometric drawing of the object appears with a rectangular grid and the starting corners (point A, in Figure 1a). The student through the use of joystick can control the cursor to draw the appropriate lines. At any time the students can add or delect lines (Figure 1b). The deleted line blinks until the figure is redisplayed. The rubber band cursor shown in the figure aids in measurement and drawing. When the student is finished he/she can check the answer by using the menu option of check answers (Figure 1c). The software will supply the correct solution and blink the incorrect lines and highlight the missing lines. Thus the students get an immediate feedback on errors and they can redo the problem if required to get a hard copy to submit to the instructor.

On the other hand if the student selected ISO DRAW and the assigned problem. The three principal views of the object appear in the lower left hand corner of the screen and an isometric grid appears on the rest of the screen with the starting corner (point A in Figure 2a). The student once again moves the cursor using the joystick and draws the lines. He/she can delete lines at any time or enter them if they think they are correct. The line to be deleted is left blinking until the redisplay of the figure (Figure 2b) the rubber band cursor is very helpful in drawing diagonal lines. The answer can be checked by selecting check answer on the menu when the student thinks he/she is finished (Figure 2c). The hard copy of the answers can be obtained for submission if so desired. The software once again will blink incorrect lines and highlight the missing one.

CLASSROOM USE OF SOFTWARE

Each student enrolled in a beginning course in engineering graphics was provided with a sheet which provided the logon procedures and step by step instruction on how to access the assigned problems. Each menu item was also explained on the sheet. The students were taken to the workstation sites by the instructor and the use of the workstation was demonstrated so that students knew exactly what to do. Beyond that the students could access the workstation at all the times the computing facilities were open (16-18 hrs/day).

The instructor assigned practice exercises and selected not to grade these (since the students already had the solution). If students were having difficulty the instructor assigned additional exercises as seen appropriate. Since the software does not support classroom management concepts it was not possible to use this as a testing vehicle.

The practice in the class was traditionally to grade every drawing submitted. this was so, because it was felt that the student had to learn of his errors as soon as possible. This entailed that the instructor had somewhere between 120 to 240 object drawings to grade every time. The tutorials as presented here decreased this grading burden since the student learned of his errors immediately and the instructor did not have to feel guilty about falling behind on his/her grading or not grading every drawing.

The students were very positive about the use of software and generally felt that the immediate feedback improved their understanding of the topic. Their only complaint was that they would have liked to have this workstation available at home so they did not have to walk to the

computer facility!

CONCLUSION

In this paper a tutorial software concept for teaching traditional orthographic projection and isometric drawing concepts is presented. The software was found to be useful and helpful. In educational realm the software is of drill and practice type rather than a computer aided instruction module in the true sense. The instructor, once he/she introduces the topic to the class in a traditional manner, does not have to spend a great deal of time giving examples. Rather the students could work on practice exercises on the work station and the instructor can act as a resource peson to help answer student's question or provide a deeper understanding of the topic through personal interaction.

ACKNOWLEDGEMENT

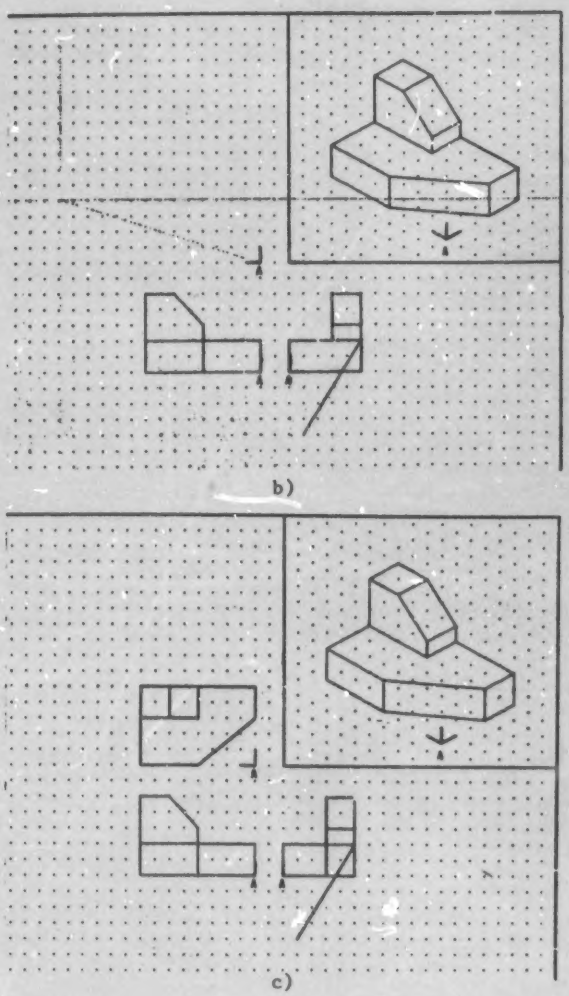
This research was partially supported by the IBM Corporation and The Ohio State University. The concept of this software was originated by Dr. Jocob J. Wolf III, formerly at The Ohio State University. The program was written by Mr. Jon Levy, an undergraduate student in Electrical Engineering at The Ohio State University.

REFERENCE

Earle J. H. Engineering Design Graphics. 4th Ed., Ma: Addison Wesley, 1983.

Hopkins M.R. Engineering Graphics Software at OSU, IBM/ACIS Conference digest 1983.

* Formerly at The Ohio State University



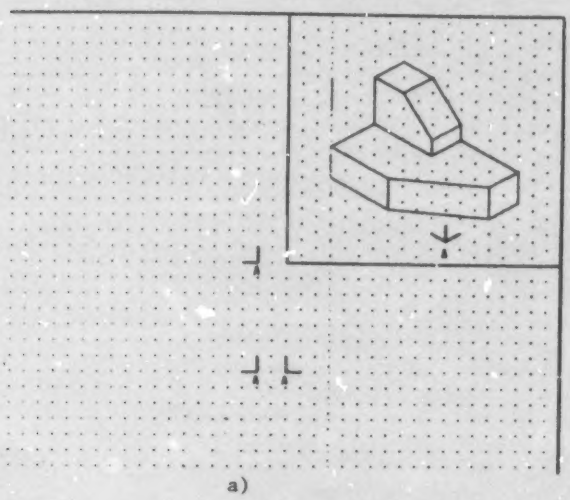


Figure 1. Orthographic Projection drawing from the isometric drawing

a) The problem as presented

b) The student enters his solution using the rubber band cursor (with errors)

c) The check answer option (missing liens are highlighted, incorrect lines blink).

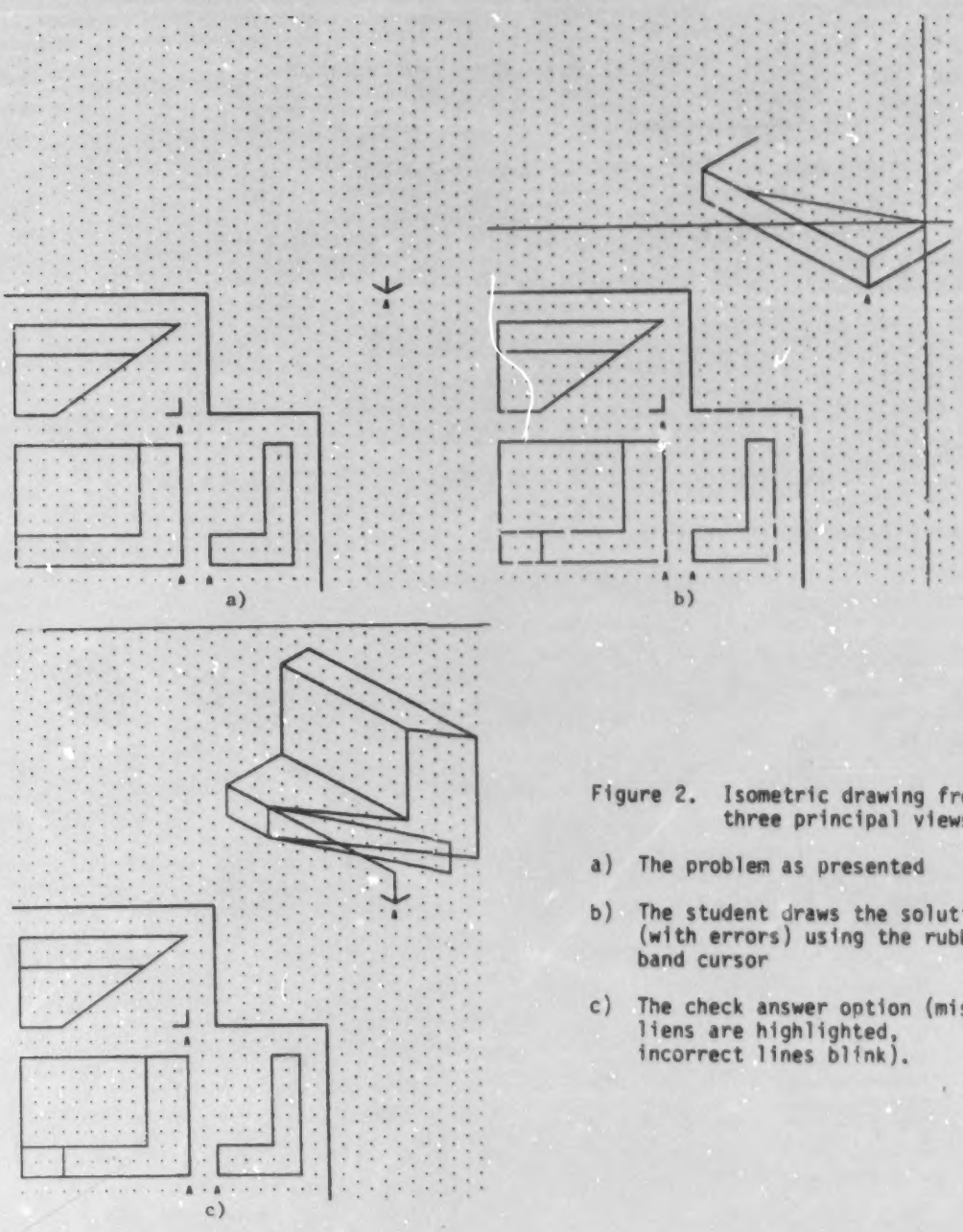


Figure 2. Isometric drawing from the three principal views

a) The problem as presented

b) The student draws the solution (with errors) using the rubber band cursor

c) The check answer option (missing liens are highlighted, incorrect lines blink).

EFFECTIVENESS OF PHOENIX AS A COMPUTER BASED EDUCATION SYSTEM

D. L. JELDEN - UNIVERSITY OF NORTHERN COLORADO

Abstract

This paper summarizes a research study completed for the PHOENIX Computer Based Education System in a Teacher Education Program. Information on the perceived value of CBE by students and achievement gains contributed by CBE are reported. Brief descriptions of time, terminal availability, lesson format, screen displays and learning outcomes are discussed. Both tutorial and testing programs were involved in the study. Information on the physiological and psychological factors related to CBE were obtained from the 77 students in an educational technology and instructional design program for teachers.

Introduction

The utilization of computers in the Educational System is being accepted by most educators. There is still some doubt, however, on the educational effectiveness of such instruction. This paper is a preliminary report of a staff study being done at the University of Northern Colorado using PHOENIX, A mainframe delivery system, distributed by Goal Systems, Inc. of Columbus, Ohio.

Although several types of computer application are found in the field of education, (simulation, drill and practice, games, tutorial and testing, etc.), this study deals specifically with the instructional tutorial and a computer based test generation system related to those tutorials. This is a preliminary report of a staff study covering an academic year of coursework. Completion of the final study is scheduled for the end of summer of 1985.

Data in this report were obtained from a pilot study during the summer of 1984 from two classes with a total enrollment of 77 students. An on-going study is underway which will add to this data base with an estimated 1000 students over 11 courses in 4 subject matter areas.

In the research process for the pilot study, no attempt was made to create or establish experimental and control groups or to objectively evaluate the academic capabilities of those students participating in the study. The preliminary pilot study was descriptive in nature designed to validate the tutorials and to evaluate the data collection system. The information was obtained from archive files created by the computer as students took lessons and tests. Demographic and attitudinal data were obtained from an evaluation form designed to look at the human factors of computer based education. Separate data exists for each course and for each program (CAI instruction and CMI tests files). This report is an accumulation of data from two courses in two types of computer application.

Population of the study

The persons involved in the pilot study were graduate students in Educa-

tional Technology at the University of Northern Colorado. The population consisted of 36 male and 41 female students. Approximately ninety percent were classroom teachers or educational administrators. The rest were full time graduate students, librarians or persons interested in computer operation or application in the public schools. The average age was 35.4 years.

The program

Two courses were used in the pilot study. The first was a Digital/microprocessor theory/lab course consisting of 8 CAI tutorial lessons and parallel test files. The second was an Introductory Computer Based Education course which consisted of 9 CAI tutorial lessons and parallel test files. The digital course covered information from Basic Gates to microprocessor systems and provided 30 contact hours of laboratory instruction and 20 contact hours of class presentation or discussion. The CBE class included a brief presentation of material on computer structure, types of computer programming language and a brief overview of the basic language. This class was primarily a 30 contact hour lecture class with open labs provided for those who wished to have practice time using the microcomputers available. In both classes, the CAI tutorial lessons were used as an adjunct to the traditional lecture and were designed to be either used as supplemental or initial contact presentations, which ever the student chose. In all cases, for the pilot study the students were given the option of testing prior to or after instruction. In the actual study, pre-testing was mandatory prior to instruction so net achievement/and instruction gain/time could be computed.

Findings of the study

Net worth or value: the first judgement made about the pilot study was that both CAI (instruction) and CMI (testing) were very beneficial to learning the materials. Eighty-six percent indicated that the CMI test and CAI instruction were very worthwhile and contributed much to their learning. Although some students indicated being a bit uneasy with the PHOENIX terminals at first over 88% said it was easy to learn and presented them with no problems. This supported previous research regarding hardware operation and student attitudes toward the computer. (Kulik, 1983A, 1983B).

Time on the System

The average lesson was designed to take about 45 minutes to complete and the average test 15 minutes. When asked to estimate how much time they spent on each lesson and test, the students agreed with the time factor. What was not expected, however, was the number of students who took the tests and lessons a number of times. Most doubled the estimated time by repeating the programs until they had mastered the material. The range was from one hour per lesson to a maximum of four hours per lesson. By not restructing access time to the system for either the lessons or the tests, a user back-up did occur. Most students complained about not having access to the terminal when they wanted it. By not scheduling student times, freedom of system access doubled estimated system usage on the average for each student.

Lesson Format and Screen Displays

The screens for the tutorial lessons were restricted to 40-45 character width with 2-3 paragraph limitations. The narrow display (1/2 the available space) made reading easier by limiting eye movement and provided for more student interaction from the terminal. Almost without exception, the students thought the narrow screen layout, use of highlighting and frequent response requirements helped maintain interest in the program and thus improve its effectiveness. Although sound, graphics and animation was not available for PHOENIX in this study, students still agreed on the usefulness and value of the lessons without those enhancements.

Each lesson was structured in accordance with the PHOENIX format. Lesson titles descriptions, objectives, clarifications and instructional items were organized for presentation. The educational model followed (Jelden, 1981, 1982) was

judged very helpful and provided for remedial as well as accelerated learning. The pacing of the lesson was determined by the operator from the keyboard. When asked if the system of presentation was any better than a text book for learning, 86% said it was very beneficial and judged it better than a traditional text.

Learning Outcomes

The goal of any educational process is to expand the students knowledge skill and attitude in a given area. As part of the pilot study, students were asked to judge several things about the system and how it contributed toward their learning experience. Ninety-seven percent of those in the study rated the computer based education program as very beneficial or contributing factor in achieving the objectives for the course. When asked if the CAI lessons helped them remember the material better than reading it in a text or listening in a lecture 81 percent preferred the CAI lesson. In reading the comments of students, many indicated that interacting with the lesson and being forced to make a decision within the lesson was the contributing factor. This supports the educational concept that students learn and remember material better in an active rather than passive mode (Andresky, 1983; Ort[illegible], 1983). The ability to decide on an item and then recieve either remediation for a wrong answer or praise and support for a correct answer seemed to make learning activities more meaningful and thus more productive. Eighty-one percent of the students in the study indicated that the felt the PHOENIX CBE system helped them learn more material than would have been possible with traditional instruction. Nine percent said it was about the same as traditional instruction. Not one indicated the CBE was a detriment to learning or that is held them back. When asked if the system should be kept and included as part of the instruction program, 99 percent said that it should be retained and incorporated as much as feasible in all classes with 69 percent saying it should be required for all students. This is compatible with existing research findings (Broussard, 1983; Baldwin, 1983; and Metrics Associates, 1981).

Testing

Because testing can be performed on individual demand by the students with computer managed instruction, they were asked to list their preferences for test administration. Sixteen percent opted for teacher administered tests at the end of the quarter, 56 percent asked to have tests available on student demand and 18 percent voiced no opinion and indicated that either would be OK. In general 97 percent of the students in the study said the PHOENIX CBE system was very beneficial and rated the option of taking tests several times a contributing factor to their improved learning. It should be pointed out here that the system contained twice as many test items in the data file than was generated on each students exam. Also, test items we randomly presented to the students from this file by objective and difficulty level. This meant that no student recieved the same test on any retake. The record for test tries in the study was an average of 13 per lesson, with one student taking 110 tests over the 8 lesson series. The class average was 3.8 or about four tries per lesson to achieve mastery which was set at 85%. It should be noted that one try was a pre-test which was used to evaluate student performance prior to instruction. Thus, students averaged about three post tests after instruction to achieve mastery or satisfaction from their point of view. If students did reach the mastery level of 85% on the first Post-Test try, they could continue to take the tests as many times as they wished if they wanted to raise that mastery level toward 100%, which many did. With the time threat removed, students were less apprehensive about the testing process. (Kulik, 1983C).

Physiological Factors

Students were asked to respond to certain items in an attempt to evaluate the physical effects which CBE might create, such things as eye strain, the environment, time of study and response time were evaluated in this manner.

Thirty-nine percent of the students did indicate some eye fatigue during the testing/instruction

period. Comments from several students indicated this was a problem only after prolonged periods at the terminal. A session of from 30 to 45 minutes exposure was not detrimental and posed no major problem. Screen reflection from back or overhead lights did interfere with the ease of reading for materials displayed on the video screen. One interesting point made by the students was that by using PHOENIX, 43 percent felt they spent more time studying the course than they would have if it were presented in the traditional manner. Although no attempt was made to measure the time/mastery function of CBE in this study, students did raise their level of competence as measured by objective test scores administered at the end of the course. Also, 37 percent said they spent less time studying when they used PHOENIX and still reached a level of competence beyond that expected with traditional instruction. Basically, the utilization of CBE makes provision for both the rapid and remedial learner and can provide educationally sound and meaningful experiences for students at different aptitude levels. The time spend by each student to attain the prescribed levels of competence is determined by the students rather than clock/credit hours. Other research supports the reduction in learning time with CBE (Burns, 1983; Myers, 1983; and Meyer, 1983).

Psychological Factors

The last major phase of the study dealt with the attitude of students toward the CBE system and toward learning on the system. Students were asked to rate their feelings toward CBE and the effective use of their time. Ninety-seven percent indicated that CBE was good utilization of their time and that time spent on the system was more productive that spent in traditional instruction. Seventy-eight percent felt the personalization of the programs, the records and reports generated on their status as they moved through the materials was very good. Some concern was raised by students who said at some time during the course, they did feel a bit threatened by the computer. Forty-five percent had this feeling at sometime during the quarter but by the quarters end, 95 percent said that it was no major problem to them in achieving the course objectives. Related to this, 95 percent of the respondents said they liked to take lessons and tests on the system and felt good about the educational worth of the experience. This was also supported by previous research (Davis, 1982).

In the open comment section of the evaluation instrument, students provided the following positive and negative observations of the system. These following observations are rated in priority by frequency of occurrence.

Positive Observations

- Less anxiety toward learning --- more fun
- PHOENIX enhanced my learning experience
- Enjoyed learning the content more with PHOENIX
- The CAI prepared me well for the exams
- Liked the control of pacing my own learning experience
- Removed the guesswork from learning by providing objective specific objectives
- Liked the ability to upgrade tests with the opportunity for many sessions
- Excellent system for reviewing course material once instruction was given

Negative Observations

- Terminal availability was a problem
- Response time was too long in some instances (computer load was heavy)
- Had some eye problem in extended sessions

Conclusions

Based on the results of the pilot study and from the records housed in the computer data bank for lesson and test files, the following conclusions about the instruction/test programs and PHOENIX can be made.

1. Students spent an average of 16 minutes per lesson test and 90 minutes per lesson tutorial.

2. The student to terminal ratio of 12:1 was sufficient for most CAI/CMI applications. This is based on an 8 lesson, 10 week structure with lessons and test averaging about 1.75 to 2.00 hours per unit.

3. Graduate level and adult learners, both men and women, did well in achieving the course objectives and had a positive attitude toward computer basic learning.

4. The CBE system worked equally well for teaching basic content information in lab type activity courses and for traditional lecture type courses.

5. Computer based education was judged very beneficial when used as an ajunct or support system to traditional instruction.

6. Student control over pacing of the lesson and time spent in the lessons is crucial to the development of a positive attitude toward C.B.E.

7. Keeping screen test displays reasonably narrow (40-50 characters) and limiting the amount of information displayed to single concepts or statements provided for more interaction between the system and student thus holding student interest and improving learning.

8. For adult learners, instructional lessons which were generated according to some logical validated educational model proved to be effective. Preparing quality CAI materials is a complex task but can be learned in a reasonable period of time.

9. The PHOENIX CBE system is an effective delivery system when judged by the students attitudes toward it and as evaluated by final exam test scores over the objectives of the course.

The utilization of the computer as an interactive instruction system in higher education has support both from previous research and from the preliminary results of this study. Discussion with students in the courses involved did raise some social/psychological questions regarding CAI delivery. Some indicated they were programmed rather than taught. This issue has also been raised in other publications (Kiesler, 1984) and will need to be addressed in future research. As of now, CBE is taking its place as an accepted mode of instruction which appears to work best as an adjunct to traditional instruction systems.

References

Andresky, Carl J., (1983). "An analysis of the cost and effectiveness of utilizing computer-assisted instruction to teach library skills at the intermediate level," unpublished doctoral dissertation, University of Pittsburgh.

Baldwyn Separate School District, (1983)."A Gifted Model Designed for Gifted Students in a Small, Rural High School," Baldwin Miss, 1980-1983.

Broussard, Rollard L. (1983). Broussard, Rolland L. "Homebased Computer Assisted Adult Education Project--Phase III," Final Project Report, Louisiana State Department of Education, Baton, Rouge, September 1, 1982 through August 31, 1983, BBB14696.

Burns, Monte B. and Swearingen, Laura, (1983). "A justification for the use of CAI with Physically Handicapped," CEC National Conference, March 10-12, 1983, Hartford, CT, Conference Procedings.

Davis, Brian Lee (1982). "The Effectiveness of Computer Assisted Instruction and Behavior modeling for performance appraisal training," Unpublished Doctoral Dissertation, University of Minnesota.

Jelden, D. L. (1981). "A CAI Course-writer system for the microcomputer" AEDS Journal, 14, 3, 159-168.

Jelden, D. L. & Brown, Bill R. (1982). "A Generalized Learner Controlled Education System," Journal of Educational Technology Systems, 11, 1, 3-21.

Kiesler, Sara, Siegel, Jane and McGuire, Timothy W., (1984). "Social Psychological Aspects of Computer-Mediated Communication," American Psychologist, 1123-1134.

Kulik, James A., et.al., (1983). "Effects of computer-based Teaching on Secondary School Studens," Journal of Educational Psychology, 75, 1.

Kulik, James A., (1983B). "Synthesis of research on Computer-Based Instruction," Educational Leadership, 41, 1, 19-21.

Kulik, James A., (1983c). "Synthesis of Research on Computer-Based Instruction," Educational Leadership Magazine, 41, 1, 19-21.

Ortman, Lois Nemtwich (1983). "The Effectiveness of Supplementary Computer-Assisted Instruction in Reading at the 4-6 Grade Level," Unpublished Doctoral Dissertation, University of the Pacific.

Metrics Associates, Inc. (1981). Evaluation of the Computer Assisted Instruction Project, Cheimsford, Mass.

Meyer, Jerome E. (1983). "Computer-Based Auto Mechanics Training," Industrial Education Magazine, 72, 7, 22-45.

Myers, Michael M., (1983). "Remedial Education in College: The problem of unprepared students," Southern Regional Education Board, Atlanta, GA.

COMPUTER BASED TEACHER TRAINING

G. Phillip Cartwright and Cynthia N. Schloss

The Pennsylvania State University

Abstract

In addition to the main campus, Penn State maintains 17 other campuses through which freshmen and sophomores take the first two years of most undergraduate degrees. Some majors, though, do not have faculty at all campuses and students must wait until their junior years to take the first courses in their majors. In an attempt to alleviate this problem, the first course in Special Education is now available at the branch campuses through the use of microcomputer based instruction. The 3 credit course is taught entirely by microcomputer and provides the material necessary for all education majors to pass a State mandated exam related to educating the handicapped. The course demonstrates an efficient and flexible use of technology in higher education and is in use in a dozen locations in the United States and Canada.

Problem 1

The Pennsylvania State University is a multi-campus institution with 17 Commonwealth campuses in addition to the main campus. Most students attend a branch campus for two years then transfer to the main campus for the last two years of their baccalaureate programs. Our problem: there are no Special Education faculty at the branch campuses but there is a real need to begin instruction about problems of educating the handicapped before the junior year of a 4-year program.

Problem 2

In 1980, the State Department of Education mandated that all persons who apply for teaching and instructional certificates in any area of education (e.g., counseling, teaching chemistry, supervision, elementary education) must demonstrate ten generic competencies dealing with education of the handicapped. A similar problem existed: insufficient staff at the branch campuses and the main campus to provide even modestly individualized programs or courses of study for such diverse groups. The ten generic competencies are:

1. UNDERSTANDS THE LEGAL BASIS FOR EDUCATING STUDENTS WITH HANDICAPS IN THE LEAST RESTRICTIVE ENVIRONMENT.
2. UNDERSTANDS THE IMPLICATIONS WHICH HANDICAPPING CONDITIONS HAVE FOR THE LEARNING PROCESS.
3. RECOGNIZES STUDENTS WHO MAY BE IN NEED OF SPECIAL SERVICES.
4. MAKES USE OF APPROPRIATE RESOURCE AND SUPPORT SERVICES.
5. CONFERS WITH AND REPORTS TO PARENTS ON EDUCATIONAL PROGRAMS FOR STUDENTS WITH HANDICAPS.
6. FACILITATES THE SOCIAL ACCEPTANCE OF PERSONS WITH HANDICAPS BY ENCOURAGING POSITIVE INTERPERSONAL RELATIONSHIPS.
7. USES INDIVIDUAL, GROUP, AND CLASSROOM MANAGEMENT TECHNIQUES FOR EFFECTIVE ACCOMMODATION OF STUDENTS WITH HANDICAPS.
8. ASSESSES THE EDUCATIONAL NEEDS OF STUDENTS WITH HANDICAPS.

9. MODIFIES INSTRUCTIONAL STRATEGIES TO PROVIDE FOR THE INDIVIDUAL NEEDS OF STUDENTS WITH HANDICAPS.

10. EVALUATES CLASSROOM PROGRESS OF STUDENTS WITH HANDICAPS.

Problem 3

There is a need to provide a high quality computer based correspondence course to teachers and parents who want basic information about handicapped children.

Our Solution

In response to the needs sketched above, the University has developed a series of microcomputer modules and an entire three credit course to assist University students, parents, and inservice teachers to obtain basic information about the handicapped. An innovative microcomputer based procedure was developed to evaluate the extent to which students and inservice teachers have achieved the competencies and prescribe remedial strategies if necessary. The quizzes and modules were tested, are in use at the University, and available for use by other institutions.

Now, instead of waiting until their junior year to obtain instruction dealing with education of the handicapped, students at any of the campuses can take a three credit course in Special Education at any time. Similarly, students, parents, or teachers in rural areas can take a full course or a set of modules on their own if they have access to a computer.

An Introduction to the Education of the Handicapped is given by microcomputer and is a "stand-alone" course; all instruction is given by the computer through a series of computer modules. The modules are easily transportable to other institutions and use the following computers: Apple (II+, IIc, IIe); TRS-80 (Models III and IV); IBM (PC, PC XT, PCjr).

The course, or a selected set of modules prescribed for the individual student, also prepares students for the special education generic competency examination, which is required University-wide.

The full length course titled EDUCATING SPECIAL LEARNERS is a 3 semester hour course and deals with the problems of educating children who are handicapped, or gifted, or who have other special problems. The course covers procedures for assessment of youngsters, litigation and legislation, definitions, characteristics, placement, generic teaching strategies, and coordination of services. It is a stand alone, computer assisted instruction course, which can be taken through Penn State's programs in resident instruction, continuing education, or correspondence. At Penn State, three semester credits are awarded for successful completion of the course.

The computer assisted instruction course requires no knowledge of computers by the student or instructor. An introductory disk plus other material fully explains what is required of the student. Each student is expected to purchase a textbook and a reference manual. The student reads a chapter in the textbook before sitting down at the computer. Then, the student merely inserts the first disk of the first chapter in the disk drive and works through the computer chapter on an interactive basis. Each chapter is contained on from one to three computer disks. The manual accompanying the disks provides additional information, off-line graphics, and related material. In addition to completing the 18 chapters on the computer, the student takes periodic quizzes. Students are on their own to pursue the materials and may do so at their own pace. However, students should plan to schedule 3 to 5 hours per week computer time. Each of the 18 disks takes 60 to 120 minutes to complete, and the instruction time spent on the computer equals roughly 37½ clock hours, the same amount of time as is required for a conventional three credit course. Because the program is self-paced, some students may complete the lessons quickly, where others may work at a slower pace and repeat the lessons as often as they wish.

The total course package includes the following:

Set of 19 double sided disks.

Reference Manual, 195 pages. Objectives, descriptive outlines, and graphics provided for each of the 18 chapters.

Instructor Manual, 101 pages. Provides the instructor of the course with information required to monitor the course. Also multiple choice quizes are provided for each of the chapters.

Proctor Manual, 18 pages. Provides information to oversee the mechanical or technical operations of the course.

Monitoring of the course is a very straightforward affair and is being conducted on other campuses by mathematics, English, educational psychology and other nonspecial education personnel. So far we are very pleased with the results and have had no significant problems with the monitoring. The course offering costs are considerably lower than a conventional course. The course can be offered for as few as one or two students or as many as your equipment and scheduling will permit.

The course has been in use at Penn State for 2 years; approximately 300 students have completed the course. The entire program is available from Penn State's Department of Audio-Visual Services. Several universities and school systems in the US and Canada have obtained the material for use within their own institutions.

Model for Delivery of Braille Instruction on a Statewide Basis

Marie Knowlton, Ph.D.

University of Minnesota

ABSTRACT

The problem of providing adequate training in braille transcribing to personnel in dispersed rural areas is addressed. A pilot project was initiated at the University of Minnesota to teach braille transcribing using microcomputers and braille text editor software programs. The results revealed an increased efficiency in learning braille while maintaining ability to transcribe by braille writer. The projected development of this project which utilizes intercommunication between a main frame computer and microcomputers throughout the state is discussed.

Model for Delivery of Braille Instruction on Statewide Basis

Currently there is a nationwide need for teachers of special education trained to teach visually handicapped students who are being educated in their local schools. The needs are greatest in areas that are more rural and these areas often have difficulty attracting and maintaining the services of qualified personnel. Forty-four institutions of higher learning offer instruction towards licensure as a teacher of visually handicapped students but the need remains to employ these graduates and retain them in the rural areas. In the state of Minnesota, our approach is to identify school districts in the state that have current or projected needs for teachers of visually handicapped. It is our intent to attract interested and qualified people already integrated in these communities and then to tailor the delivery of instruction toward licensure to these individuals. One of the areas of instruction is literary and mathematical braille transcription.

In recent years there has been a renaissance in the instruction and use of braille. This is due to (1) the development of a comprehensive code for mathematics (Nemeth Code) which was formally adopted in 1968 with later revisions in 1972; and (2) the invention of suitable braille printers that could be interfaced with microcomputers, granting access to computer technology by the braille reader. The re-establishment of the importance of braille as a notation system necessitates that the teachers of visually handicapped students be competent in transcribing literary and mathematical braille. At the University of Minnesota more than 25 per cent of the formal instruction toward licensure as teacher of visually handicapped is devoted in instruction in braille.

BRAILLE TRANSCRIBING

The traditional methods of teaching braille have been by use of a manual six-key braillewriter which operates somewhat like a simplified typewriter and embosses heavyweight paper with the appropriate configurations of the braille symbols. All braille is composed of a system of dots in a two-by-three matrix which constitutes the braille cell. One to six dots complete a symbol allowing a total of 63 signs. (The empty cell is equivalent to a space in print.) Each key on the braillewriter embosses one of the six dot positions in the cell. However, braille is not a letter for letter transcription of print material. Most braille is transcribed in what is known as Grade 2 braille, a shorthand contracted form which utilizes contractions, short form spellings, and symbols that do not exist in print English. In addition, the mathematical languages (algebra, geometry, symbolic logic, etc.) require the use of many more symbols. The 63 possible braille configurations must be utilized to serve the purposes of all these numerous symbols. This is accomplished much in the same way that a control or escape key is used in computer programs. A numeral indicator, letter indicator, or dot 5 indicator each causes the following symbol to be interpreted by the reader in a very different way.

Because braille is so complex a writing system it requires commitment and effort to master. The mechanical method of producing braille by braillewriter which directly embosses the page also contributes to its difficulty. Once a configuration has been embossed, the paper itself has been deformed and any subsequent correction of the embossed cell is both time consuming and inadequate to return the page to its original condition. There will always be some indication of the previous symbol on the page. The only absolute way of eliminating an error in transcription has been to rebraille the complete page. It is inefficient and of minimal value for the student who has correctly embossed nearly 1,000 cells/page correctly to rebraille a whole page.

A fairly recent advent in the transcription of braille has been the development of text editor programs for computers. These programs utilize a six key input which simulates the braillewriter keyboard. Input is stored on data files which can then be manipulated in a manner similar to most word processing systems. Deletions, insertions, and corrections become part of the transcriber's repertoire. The display on the video terminal can be either in a dot matrix display which simulates embossed braille or in ASCII equivalents, of the 63 symbols.

THE PILOT PROJECT

During a five week summer session of 1984 at the University of Minnesota, a pilot project was initiated to test the feasibility of using text editors in conjunction with the more traditional means of teaching braille writing. Three students were provided with traditional braillewriters the first class session. The initial lesson assigned was the task of learning the alphabet, capitalization

rules, and numerals. This is a fairly easy task and permits one to transcribe most literary material into a simplified form of braille known as Grade 1 braille. In the second class session the students were given access to a word processing program, either ED-IT (Stepp, 1982) or BRAILLE-EDIT (Holladay, 1984) and taught how to access it. Both of these programs run on several different microcomputers but all members of the class used APPLE IIe's with 80 column cards for input. The students were given no further assistance with microcomputers. They were told that they were permitted to hand in assignments in hard copy by manually embossing the required material on a braillewriter or alternately, they could submit their assignments on floppy disk for evaluation. The course requirements were that at least one assignment had to be on floppy disk but that the final would be transcribed manually during a predetermined examination time. The assignments were evaluated as they were submitted. Comments were inserted in the text file in braille by re-entering the editor, calling up the text file and inserting specific error symbols after transcribing braille error. An additional blank line or lines are added before the next line of text and the evaluator inserted, a comment on the correct method of transcribing the error. These error comments start in the fifth cell, or space, from the margin. Since braille paragraphs begin in the third cell, the evaluator comments are readily identified within the text. After evaluation the assignments are returned to the students.

EVALUATION

After the second lesson students, voluntarily, submitted all remaining assignments on floppy disk. It was their personal preference although two of them had no previous experience with computers and none of them had experience with braillewriters until the first class session. They commented on the ease of error correction and the lightness of touch required on computer compared to braillewriter. The major complaint expressed was the difficulty in reading the braille dot configuration on the monitor. Because one omitted or misplaced dot can change the meaning of a whole word, accuracy is very important. The eye strain created by trying to read the available monitor could be corrected in one of several ways. There are versions of the braille editing programs that output to the monitor in larger sized configurations. Duplicate programs could be bought in these modes. An alternative would be to invest in additional hardware, either a more expensive standard monitor, a large print monitor, or a braille printer. These solutions are more expensive but would permit the use of all existing course software.

The instructor's evaluation of student performance was enthusiastic. The use of microcomputer text editors greatly improved the efficiency with which the students completed the course. Previous experience in teaching braille transcribing required six to nine months for students to reach the same level of competence. Since the final was to be transcribed manually, the ability to produce embossed braille in the traditional manner was also evaluated. Manual embossing is an important skill for the prospective teacher who will often be in situations without computer hardware to produce embossed material for students and must do it on a braillewriter. All students were efficient and competent in operating braillewriters. The speed of their output was more dependent on their cognitive ability to apply the transciption rules than on actual mechanical manipulation of the machine or additional effort required to press the keys. The major complaint by the instructor was, again, the difficulty of proofreading assignments on the available monitor. Attention is being focused on this issue as we proceed in the development of our braille instruction.

PROJECTED PROJECT EXPANSION

The pilot project described above is only the first phase of the course development. Now that we have data to demonstrate the efficiency of learning braille transcribing using microcomputers, we wish to address the issue of presenting instruction to prospective students dispersed widely over the state. Correspondence courses are one possible means of solving the problem. However, the efficiency of using microcomputers for text editing would be negated by the necessity of using the mail for exchange of assignments. Our plan is to establish an account on one of the main frames at the University.

Students will be required to have access to a microcomputer with a modem. We will supply them with a copy of the text editing program, a carrier program, and instruction modules. When an assignment is completed, the student will access the account on the main frame and upload the assignment from the microcomputer. The instructor can then access

the assignment, make corrections and return it to the account where it will be available to the student. Using this model a minimum of expensive computing time is needed. Most time will be spent locally on the microcomputer. Only attaching, up and down loading of files, and storage of files will be done on the main frame. Additional charges to the student will be variable depending on the cost of long distance telephone charges to the main frame from their area.

Using the model presented here we hope to partially address the need for qualified teachers of visually handicapped students in the areas where they are needed. That we can do this efficiently will be partly due to the efficiency of computer instruction over large geographical distances. An additional benefit will be realized in that our teachers will also gain in computer literacy while completing braille instruction.

References

Holladay, D. BRAILLE-EDIT, 1984. Available from Raised Dot Computing, 408 South Baldwin, Madison, WI 53703

Stepp, R. ED-IT, 1983. Station A, Box 5002, Champaign, IL 61820

AN INTERACTIVE MICROCOMPUTER SYSTEM FOR TEACHING READING AND WRITING TO YOUNG COMMUNICATIVELY HANDICAPPED CHILDREN

Philip M. Prinz and Keith E. Nelson

The Pennsylvania State University

Abstract

The present research involves a new method for teaching communicatively handicapped children (3-8 years) to read and "write" using a microcomputer. The system builds in perceptual salience, specially developed vocabulary, and animation of color pictures and optional graphic representations of signs from American Sign Language (ASL). The learning mechanism underlying this novel instructional system is best characterized as interactional and exploratory, reflective of the way in which children acquire a first language. Results have demonstrated a significant improvement in reading and communication skills.

Introduction

This research project involved an investigation of the effectiveness of microcomputer instruction for improving reading and writing achievement in young communicatively handicapped children. The study incorporated the use of a portable microcomputer to allow the child to initiate communication from the very onset of instruction. Traditionally, children using computers have been required to respond to statements and questions and their responses have been required to fall within a small range of acceptable answers. The approach used here instead provided a more flexible set of possibilities and gave the child a more active role.

Method

The initial research was conducted at The Pennsylvania School for the Deaf in Philadelphia. The software developed for the project included printed words and phrases, animated graphic representations of pictures to illustrate word meanings, and optional animated graphic representations of signs from American Sign Language (ASL).

Subjects

The participants in this phase of the study were 32 deaf children between the ages of 3 years, 2 months and 8 years, 5 months. They all exhibited average or better intelligence as indicated by a performance scale of an accepted intelligence test. Only subjects with a severe to profound hearing impairment were included. The child's mode or modes of communication were not a criterion for selection. An initial assessment was conducted of the child's basic communication skills including speech, sign language, fingerspelling, and simultaneous communication (speech and sign). After the child's primary mode of communication was determined, communication measures were administered in that mode. These measures included a receptive and productive vocabulary test at the single word/sign level, a sentence imitation test with items which increased in terms of length and grammatical complexity, a measure to evaluate the comprehension of grammatic and semantic relationships and a referential communication task to assess the child's ability to convey information to another communicator. Additionally, baseline measures of the child's reading abilities were completed. Based on this information specialized programs of instruction were developed for the children.

Equipment

The main equipment used in the project is the Apple II microcomputer system (+48K bytes) which is set up with a special, large interface keyboard with interchangeable overlays allowing more flexibility than a regular computer keyboard. This special keyboard, the Express-1 (Prentke Romich Company) and the Keyport 717 (Polytel Corporation), provides interchangeable overlays which allow whole words (or pictures) to appear directly on the keyboard in sizes appropriate for beginning readers, and which are easy for the child to activate. When the child makes a selection of a picture or sign, the corresponding word or phrase as programmed is printed on a printer and displayed on a large color monitor connected to the Apple II microcomputer unit.

Courseware

As part of the current phase of the research, a software package--the ALPHA Interactive Language Series--was developed. The ALPHA program consists of different lessons designed to assist communicatively handicapped students who are just beginning to read. Whole-word overlays for the student interface keyboard correspond to each ALPHA lesson. The focus of the ALPHA series is to develop a basic reading and "writing" vocabulary, and the ability to create simple sentences using this vocabulary. In the ALPHA programs, the computer displays items in the form of printed words, animated pictures, and optional animated graphic representations of manual signs.

Over a period of 9 months, each child received between two and five 10-15 minute computer instructional sessions weekly. The set of words, pictures, and graphics employed were gradually introduced. When the child pressed the appropriate word or sequence of words, the computer responded accordingly. For example, suppose the child pressed the keys for "horse jumps ball." At this point, the video monitor would typically display both the printed words, and redundant pictorial information. Then the same sentence could be repeated with animated signs on the TV screen. In addition, the teacher had the option of requesting more information or commenting on the child's message. This could be done, for example, by asking the child if she/he wants to see a big horse or a small horse or by telling the child something else about a horse--again with display of appropriate print and graphics. This served to create an atmosphere where the child actually initiated and used printed information in an active, interactional and pragmatic manner to exchange messages with another person.

The child's oral and manual skills (and any writing/printing skills) were also tapped by the teacher in the process of instruction, so that the child learned to communicate about the computer-assisted material and learned to flexibly extend the new reading knowledge and vocabulary beyond the computer.

The children's progress in the use of the novel system was formally evaluated both from the printers attached to the Apple II and the student keyboard and results of specific language measures administered. The exact records of child key presses and teacher key presses were available for analysis.

After the child explored the vocabulary and had created a variety of sentences in a lesson, the experimenter switched to the computer test format to assess the child's level of competence in producing words and sentences. The computer generates an animated pictorial depiction of individual nouns or noun-verb-noun sequences randomly selected from the vocabulary in the ALPHA lesson. The animation is displayed pictorially on the TV screen and the child then selects the corresponding word or sentence on the student keyboard.

Results

The results showed that the children with better preinstructional sentence imitation scores ($>.60$) and those who received more intensive computer instruction (21 or more sessions) performed at a higher level on the Computer-Generated Word and Sentence Tests.

Secondly, a Generalized Reading Vocabulary Test was given to a subgroup of 13 subjects between the ages of 3 and 8 years, randomly selected from the total sample. This measure consisted of a series of individual words and sentences printed on flash cards which the child was required to read. A pretest was given in one type font and a posttest in a different font. The overall mean percentage of improvement was 86.7%.

Because it was hypothesized that general communication skills could possibly be facilitated by the interactive approach to teaching writing and reading used in this study, the subjects were administered a Sentence Imitation Test prior to and following microcomputer instruction. This test measured imitative responses in the children's best communicative mode (i.e., ASL or simultaneous communication). The results indicated that children receiving more intensive computer instruction demonstrated greater improvement on the Sentence Imitation Test. The overall mean percentage of improvement for younger children (3-5 years) was 31.8% and for the older group (6-8 years) 8.3%. The combined mean percentage of improvement for both age groups was 15.9%.

Discussion

Initial results using this newly developed interactive approach to reading and writing instruction with the microcomputer is beginning to demonstrate that deaf children as young as 3 years of age can acquire reading and computer "writing" skills. Also, preliminary results are showing that this computerized instructional approach has an impact on stimulating general communication skills. Extensions of the present exploratory work will involve internal comparisons concerning the optimal presentations of the signs, pictures, text, and teacher input for facilitating the children's processing of the material and their acquisition of new reading, writing, and general communication skills.

In sum, at the theoretical level as the reserach is extended and refined, it will be possible to learn much about how different input combinations are used by deaf children in the acquisition of new reading and writing skills and new skills in using language in nontext modes.

This research should contribute to an understanding of the extent to which written English vocabulary and sign vocabulary are coded categorically in short-term and long-term memory by deaf children. Also, it should provide important information relative to the extent to which textual material is translated into a manual-based code for storage in memory. Future research should focus on variations in cognitive styles, information-processing and individual differences in relation to reading, writing and general communication.

Further research utilizing a modified version of the ALPHA program is being conducted with other groups of communicatively handicapped children (e.g., mentally retarded and multiply handicapped). The overall significance of this research lies not only in testing a new computer-assisted instructional approach to reading and writing in communicatively handicapped children but also in the later educational options that would be opened by early success in written language instruction. Research is attempting to minimize, through a new form of early interactive microcomputer training, many of the adverse social and educational consequences too often associated with communicative deficits.

Recommended Readings

Geoffrion, L. D., & Geoffrion, O. P. (1983). Computers and reading instruction. Reading, MA: Addison-Wesley.

Mason, G. E., Blanchard, J. S., & Daniel, D. B. (1982). Computer applications in reading (2nd ed.). Newark, DE: International Reading Association.

Prinz, P. M., Nelson, K. E., & Stedt, J. D. (1982). Early reading in young deaf children using microcomputer technology. American Annals of the Deaf, 127(5), 529-535.

A MICROCOMPUTER TUTORIAL OF EKG PRINCIPLES

Kenneth J. Friedman, Ph.D., New Jersey Medical School
Joseph Boyle, III, M.D., New Jersey Medical School
Joy Black, New Jersey Medical School
Gloria Robinson, New Jersey Medical School

ABSTRACT

A microcomputer program which teaches basic EKG principles has been developed for the IBM-PC microcomputer equipped with a graphics board and color monitor. The program was written in PILOT (1.4) and BASICA to utilize the best features of both languages. The program has two main divisions: Basic Principles, and Simulations. The Basic Principles portion of the program is divided into five subroutines: Electrophysiogy, Cardiac Excitation, EKG Terminology, EKG Leads, and Electric Axis. These subroutines can be accessed in any sequence from the menu. At the end of each subroutine is a series of questions which students must answer correctly in order to progress. The Simulations portion of the program contains three subroutines. The first subroutine demonstrates the relationships between P, QRS, and T loops and the scalar EKG traces for leads I, II, and III. The second subroutine permits the student to alter the size of the left and right ventricles and then calculates and displays the resultant vector loop. Finally an exercise in determining the MEA is presented. The computer randomly varies the size of the right and left ventricles and displays the resultant vector loop and standard limb lead traces.

This program heavily uses graphics, animation and interaction with the student to present the material. These are inherent characteristics of a microcomputer system which we feel can enhance student learning and comprehension.

INTRODUCTION

Although computers have been used for instructional purposes for approximately 15 years, computer based instruction (CBI) has expanded markedly since the advent of the microcomputer. Microcomputers probably have increased the popularity of CBI by making courseware development easier and giving larger numbers of students access to these materials. The ease of programming, increased cost effectiveness, and greater independence and mobility of microcomputers are factors which contribute to the rapid growth in courseware development. The increase in number of machines available for student use has promoted student utilization of these materials.

Early CBI programs consisted almost entirely of question and answer tutorials with a discussion of the student's selection (Schwartz and Hanson, 1982). Some of these tutorials have been branched so that the student could progress rapidly through familiar material or review at length if difficulty was encountered. While these programs have provided some benefits, primarily increased rate of learning (Marion, Niebahn, Petrusa and Weinholtz, 1982), they have not utilized the newer microcomputers' abilities to present graphic models and simulations. CBI incorporating these elements would permit greater student interaction, and enhance the student's understanding of complex interrelationships in the sciences. Such programs could be devloped to cover specific subject areas and then used during lecture, conference, or independent

study to demonstrate important concepts and interrelationships.

In our institution, we have developed and used microcomputer CBI in both Graduate and Medical Physiology courses. Programs incorporating graphic models and simulations have been developed for selected aspects of hemodynamics, gas exchange and oxygen delivery in the cardiovascular and respiratory sections of these courses. These programs have been well received by our students.

This paper describes the contents and concepts contributing to the development of a tutorial microcomputer program. The program presents the fundamental concepts of the electrocardiogram (EKG) and tests the students' understanding and ability to utilize these concepts. The program has been written for the IBM PC computer (DOS 2.0), requires a graphics board and a color monitor. Two computer languages, BASIC and PILOT, have been used to take advantage of the programming strengths of each of these languages.

PROGRAM DESCRIPTION

Overview of Program

The EKG microcomputer program consists of two main parts:

1. Fundamental Principles

2. Simulations

1. Fundamental Principles

This part of the program is written in PILOT and heavily utilizes graphics and animation to present the material. The program is entirely menu driven so that students can utilize the program with only a few minutes of introduction to the microcomputer, if necessary. The program is divided into subject areas which can be accessed independently from the menu, or accessed sequentially to permit students to progress through the program (Table 1). Each subject area of the program first presents fundamental principles, both textually and graphically, and then tests the student's understanding of the material with several questions. The student must answer these questions correctly in order to progress to the next subject area of the program. Unsatisfactory performance on the questions returns the program to the beginning of that subject area, thereby forcing the student to review. Since each subject area is designed to take 10 - 15 minutes to complete and can be accessed in any order by menu selection, a student may take several sessions to complete the entire program.

The first subject area of the program begins with the concepts of membrane polarization, depolarization, and dipole formation. A dipole with isopotential lines is presented to introduce the concept of vectors. Next, the conventions of Einthoven's Triangle and standard bipolar and augmented limb leads are introduced, as well as the derivation of the tri- and hexaxial reference systems from the limb leads and Einthoven's Triangle.

The program then presents the sequence of cardiac excitation and the generation of the P wave, QRS complex, and T wave. This portion of the program heavily uses graphics and animation to demonstrate the relationships between the pathway of electrical excitation in the heart and the waveforms which appear on the scalar leads of the EKG. Next, a more extensive discussion of vectors is presented in preparation for discussing methods of determining the mean electrical axis (MEA) of the QRS complex. Two methods of MEA are presented: (1) A rapid technique to determine the appropriate quadrant of the MEA, and (2) A vector addition method for more precise calculation of the MEA.

2. Simulations

This part of the program is written in BASIC since BASIC provides greater speed and better graphics commands than PILOT. A brief introduction to the concepts of a vector, a vector loop, and the scalar lead projections of vector loops is provided. A vector loop for ventricular depolarization is generated and displayed in the center of Einthoven's Triangle. Simultaneously, the QRS complexes generated by the vector loop are projected onto leads I, II, and III. The student can inspect the formation of the QRS complexes as often as desired by repeating this subroutine.

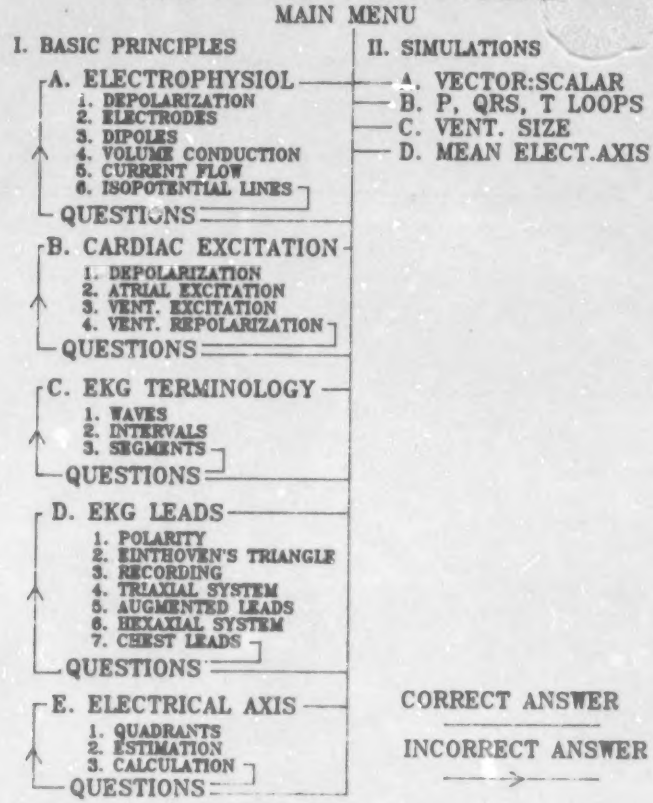
The option of varying the size of the left and right ventricles is then presented to simulate the effect, and observe the consequences, of right and

left ventricular hypertrophy on the QRS complex and the ventricular depolarization vector loop. The student is permitted to alter the size of the ventricles as many times as he/she wishes. After the presentation of each vector loop and the QRS complexes of leads I, II, and III, the program offers the option of displaying all six standard limb lead projections of the ventricular depolarization vector loop. The data for the six leads (I, II, III, aVR, aVL, and aVF) are generated in the preceding step and stored in a matrix so that all six limb lead QRS complexes can be displayed rapidly for study.

The final option of this portion of the program permits students to test their abilities to determine the mean electrical axis (MEA) from computer generated vector loops and the projections of these loops onto leads I, II, III. The student is required to estimate the MEA to within 20 degrees of the actual MEA in order to progress. If the student's estimate is more than 20 degrees in error, the computer indicates an error is present, and whether the estimate is too high or too low. Thus students are afforded the opportunity of improving and testing their understanding of ventricular depolarization vector loops, the projections of these loops onto leads I, II, III, and the calculation of MEA's from vector loops and leads I, II, and III.

Table I outlines the major divisions of the program and indicates the various paths by which students can progress through the lesson.

TABLE 1: PROGRAM OUTLINE



DISCUSSION

With the expansion of scientific information in the last decades, many science curricula have more heavily emphasized the lecture format to impart information to the students. This is especially true in the basic medical sciences where there has also been an erosion of total curriculum time devoted to these subjects (Shelesnyak, 1981). Most of the basic medical sciences have met these divergent demands by sharply reducing the assigned time for laboratory experiments and small group conferences. For example, assigned laboratory time is completely absent in physiology, pharmacology and biochemistry curricula in 10%, 40% and 65% of U.S. medical school curricula (1982 AAMC Curriculum Directory). This abrogation of laboratory time is significant since the laboratory traditionally has been the forum to teach the problem-solving (deductive) method of medical science. In addition, the laboratory provides an opportunity for students to interact in small groups or individually with faculty which provides feedback to the student regarding knowledge and performance. Due to the emphasis on lectures and the loss of laboratory time in many science curricula, the student has become largely a passive participant in the educational process.

Microcomputer technology can be used to compensate for this reduction in laboratory and small group conference time by providing students the opportunity to participate in simulation experiments. Simulation in this context is defined as the response of a mathematical model system, whether it be molecular, cellular, organ or human, to varying inputs while observing the response under both normal and abnormal conditions. Arthur C. Guyton has stated that "physiologists who have practiced computer simulation of physiological mechanisms, without exception, do not have to be sold on its importance....-when one sees the computer solution of function under both normal and abnormal conditions, one is often led into new, logical dimensions and certainly to a fuller understanding of basic physiology" (Guyton, 1980). We belive the same holds true for simulations in all of the basic medical sciences.

In the basic medical science curriculum, students learn that bodily function is dependent on a large number of complex, interrelated systems that interact in a dynamic manner during both health and disease. Learning these dynamic relationships from static graphs, formulae or pictures which are presented on slides or in textbooks encourages many students to spend all their time memorizing isolated facts rather than developing the concept that bodily function is a dynamic interaction of physiological systems. The reduction of basic science laboratory experiences in many medical schools decreases student opportunities for exploring the dynamic properties and interactions of physiological systems.

The use and availability of basic medical science simulations and graphics on microcomputer systems facilitate learning in a number of ways. Faculty will be able to illustrate dynamic relationships during lectures rather than showing static slides. This should enhance a faculty member's ability to impart information to the student. Microcomputer simulations can be used during small group conferences to enhance the student's understanding of the subject material. In addition, students will be able to study complex relationships by themselves. Perhaps most importantly, microcomputers used in conjunction with properly designed software can substitute for more costly laboratory experiments which have been deleted from courses due to insufficient funds (Visscher, 1973).

Software development is obviously critical in achieving these goals. Separate programs or subroutines should be available for lecture presentations so that the lecturer can easily sequence the proper functions to fit his/her lecture format. Software for independent student use should provide for interaction between the student and the computer so that the student can review or advance at an appropriate rate and become an active participant in the learning process. The use of graphics in such software will insure a high level of student interest.

An analysis of the literature addressing the use of computers in the basic medical sciences reveals that students using computer programs do as well, and in about 50% of the studies perform significantly better than control groups (Schwartz, 1982; Marion, 1982 and Mullen, 1980). An almost universal finding of these studies is that students using computer-based education

learn at a faster rate than do control groups. Although this increased efficiency of learning is not adequately explained, it may be due to a better focus of the student's energy and motivation on the important concepts (Mullen and Love, 1980). It may also be due to the students' becoming more actively involved in the learning process.

This program should improve the learning efficiency of students for EKG principles by the use of graphics and student interaction. The simulation subroutine visually presents the relationships between vector loops and scalar records and allows the student to alter ventricular size and observe the resultant effects. Limited evaluation by students has indicated that the program facilitates understanding of the basic concepts and principles of electrocardiography. This CBI incorporates graphics, simulations and tutorials in one teaching program, and so utilizes many of the unique capabilities of the microcomputer.

BIBLIOGRAPHY

Guyton AC: Forward in Microcomputers and Physiological Simulation. J. Randall, Addison-Wesley, Reading, MA. 1980

Marion R, Niebahn BR, Petrusa ER, Weinboltz D: Computer based instruction in basic medical science education. J. Med. Educ. 57:521-526, 1982

Mullen EM, Love RL: An evaluation on mini-computers for computer-assisted instruction in allied health curricula. J. Allied Health 9:33-40, 1980

Schwartz MW, Hanson CW: Microcomputers and computer-based instruction. J. Med. Educ. 57:303-307, 1982

Shelesnyak MC: Preface, Frontiers in the Teaching of Physiology. Edited by CS Tidball, MC Shelesnyak, American Physiological Society, Bethesda, 1981

Visscher MB: The decline in emphasis on basic medical sciences in medical school curricula. The Physiologist, 16:43-54, 1973

A Profile of Learner Traits and Learning Outcomes with Computer-Assisted Instruction

Betty J. Paulanka, R.N., Ed.D.

University of Delaware

ABSTRACT

This descriptive study examined the personality traits and instructional time factors of successful and nonsuccessful students who utilized Computer-Assisted Instruction (CAI) to learn psychopharmacological nursing. Data from this study supports the contention that there is beginning evidence to imply that there are certain individual traits that can be utilized to predict successful learning with CAI. The stepwise multiple regression analysis utilized in this study identified 17 variables that contributed to successful learning. Specific academic, cognitive, demographic and philosophical indicators emerged as the most consistent predictors of learning associated with these psychopharmacological lessons presented on the PLATO system. These results and others are explored in terms of their educational implications and suggestions for future research which can be used to plan more effective and efficient use of computers in nursing education.

PROBLEM STATEMENT

Computer-assisted instruction is a viable and equal teaching modality with advantages for the student as well as the instructor. However, the fact remains that students are unique individuals with diverse needs and learning preferences. If computers are to be accepted by higher education as a equal part of the instructional process, the computers capacity to individualize instruction to meet these needs and preferences must be developed. Successful completion of the task requires an in-depth exploration of individual traits and learning preferences in terms of which students learn best with CAI. More specifically, what is the profile of the successful learner when CAI is utilized to teach psychopharmacological nursing?

METHODOLOGY

In an attempt to ascertain the answer to this question, 109 junior nursing students, enrolled in a required pharmacology nursing course at the University of Delaware were asked to respond to a researcher prepared instrument. This tool was designed to elicit information associated with selected individual traits and learner beliefs. The PLATO system which presented the course content was programmed to collect specific data on individual students and instructional time factors. Success was measured in terms of the student's unit test score of 70 percent or better.

The PLATO program selected for this course consisted of four separate instructional units related to psychopharmacological nursing. Lessons were designed under the assumption that students had already acquired basic skills in medication administration.

Instruction began with Lesson Ia. This lesson was an introductory unit. It was the only unit without

a pretest. The primary purpose of the lesson was to assist the student in utilizing the five steps of the nursing process to assess and plan nursing actions associated with psychopharmacological nursing. Major focus was on analyzing data associated with a specific client. It was a short 15 to 20 minute lesson.

Learning objectives for Lesson II were directly related to utilizing the nursing process in the practice of psychopharmacological nursing. The objectives presented to the students on the PLATO program were as follows:

After studying this lesson, the student will be able to:

1. value the nursing process as a methodology for psychopharmacological nursing.
2. analyze client data in order to use the nursing process in psychopharmacological nursing.
3. apply appropriate steps of the nursing process in psychopharmacological nursing.

Lessons III, IV and V were directly related to the psychotrophic medications; antipsychotics, antianxiety agents and antidepressants. Pretests preceded each lesson to determine the student's entrance level knowledge base.

Students were given options to review specific knowledge where deficits occurred. Post tests were designed to measure learning and were considered to be equal to pretests.

Learning objectives for Lessons III, IV, and V are basically the same as the learning objectives for Lesson II. The only variance was in terms of the specific medication classification being presented in each lesson, i.e., Lesson III was Antipsychotic Agents, Lesson IV was Antianxiety agents, and Lesson V was Antidepressants. Lesson V varied slightly in that it was broken down into two parts: a) Antidepressants in general, and b) Lithium Carbonate, a much used antidepressant that varies significantly in its usage and method of management. Learning objectives for these lessons remain the same with a major focus on utilization of the nursing process in psychopharmacological nursing. The content in Lessons III, IV, and V was very specific. Each lesson began with some basic presentations associated with specific psychiatric disorders and appropriate medication interventions. Students were referred to specific texts for a more thorough review of disorders and specific medication classifications.

The lessons were presented as simulated clinical situations involving specific medications in association with selected psychiatric disorders. This design required the students to utilize the steps of the nursing process through constant interaction with the computer. Individual student choices determined the progression and direction of the lesson. This design emphasized application of knowledge. The well prepared student should be able to complete the three lessons within an hour and forty-five minutes.

The PLATO system of instruction allowed students to enter and leave lessons as they wished without losing their place in the lesson. Previous experience with PLATO lessons in other required courses was considered to be an asset to the student's familiarity with the system. It reduced the novelty effect for first time users.

Students were required to complete each lesson as a part of the Pharmacological Nursing course. Thus, 4 hours of class and laboratory time was allotted for this purpose. Terminals were available 16 hours a day, 7 days a week, over a six-week designated period. Students were encouraged to sign up for terminals at their own convenience. The PLATO system recorded data associated with interactions and completion of the lessons.

To reduce difficulties associated with technical problems, a computer expert was available for student assistance. Notes could be left on the system for the computer expert if she was not readily available. The touch screen option was available for students who preferred not to type.

The psychopharmacological program utilized in this study has been used in other nursing courses and was the source of data for the Boettcher, Alderson and Saccucci

Study (1980). The lessons were in the process of being revised for publication.

RESULTS

Findings in this study resulting from the Pearson R Correlation Matrix and the Chi Square Analysis suggested minimal significant associations between these selected individual traits and learning outcomes. Such findings are consistent with other research studies that have attempted to correlate within group variables and learning outcomes utilizing these less complicated statistical tests. Thus, a stepwise multiple regression model was selected to assist in the analysis of this data.

Twenty-two predictor variables, selected from the correlation matrix, were entered into the stepwise regression model with learning as the criterion variable. Of the 22 variables measured in this model, seven were considered to be significant in explaining the variance in learning for the more successful student. An F score of 3.0 or greater was required in the selection of significant variables. (SEE TABLE 1)

The seven most significant variables selected by this model in order of priority are:

1. GPI
2. Philosophical Index Score
3. Marital Status (married)
4. SAT scores
5. *Pretest scores (negative relationship)
6. Age (over 25 years)
7. Number of hours of outside employment

Ten other important variables were identified with F scores between 1.5 and 3.0 level of significance. These variables included:

8. Length of time on the PLATO System
9. Experience with medications
10. *Attitude toward lessons (negative relationship)
11. Attitude toward PLATO
12. *Pretest dates (negative relationship)
13. Amount of preparation for lessons
14. Educational accomplishments
15. General experiences

TABLE I

MULTIPLE STEPWISE REGRESSION MODEL

VARIABLE	MULTIPLE R	R SQUARE	STEPWISE RATIO(F)	BETA
GPI	0.28831	0.08026	6.54497	0.52194
PHIL. BELIEF	0.32376	0.10482	4.33236	0.29555
MARITAL STATUS	0.37320	0.13927	3.93739	0.27437
SAT	0.41876	0.17536	3.82771	-0.43686
*PRETEST SCORE	0.45602	0.20795	3.72816	-0.21375
AGE	0.47863	0.22909	3.46692	0.22092
HRS. WORK	0.49744	0.24745	3.24117	0.13379
HOURS PLATO	0.50675	0.25680	2.93699	0.00859
MEDICA	0.51258	0.26274	2.65303	-0.07287
*ATT. LESSONS	0.52098	0.27048	2.44708	-0.14880
ATT. PLATO	0.52617	0.27685	2.26224	0.06807
*PRETEST DATE	0.53118	0.28216	2.09631	-0.13201
PHARMAS	0.53653	0.28786	1.95891	0.20480
DEGREE	0.54350	0.29539	1.85660	0.08334
GEN. EXP.	0.54833	0.30066	1.74837	0.10914
CREDITS	0.55251	0.30527	1.64777	0.10265
*PSYCHIC	0.55673	0.30995	1.55889	-0.08916
CLINIC	0.55807	0.31144	1.45742	-0.04342
DATE	0.55842	0.31184	1.34942	0.02645
MOTIVATION	0.55871	0.31216	1.27069	0.02291
(CONSTANT)				

*NEGATIVE RELATIONSHIPS

16. Number of credits enrolled in
*17. Experience with psychiatric clients

* (negative relationships)

In an attempt to identify the regression model of best fit, several regression equations were used to analyze the data. Despite the use of various equations, certain variables repeatedly surfaced as significant predictors of successful learning. Thus, based on the data analysis resulting from the stepwise regression model, it can be assumed that the utilization of more sophisticated inferential statistical techniques allowed for the beginning development of a profile of individual traits that can be used to predict successful learning for this group of junior nursing students. This profile suggests:

> For the nursing student with the highest GPI; who has strong values related to academic philosophies and identity seeking ideals; who is married and has a high score on his/her SAT's but a low pretest score; who is over 25 years old and works at least 10 hours per week there seems to be significant predictive validity related to success when learning psychopharmacology with CAI.

EDUCATIONAL IMPLICATIONS

The results of this study imply that certain individual traits should be considered when planning programs or assigning students to computer assisted instruction. If the goal of education is to individualize teaching in order to meet the unique needs of the great masses of learners, then CAI must be considered as a realistic choice in terms of instructional modalities. However, as educators attempt to capitalize on this important capacity of the computer, they must give more attention to the individual traits of the learner when planning the content and the format of these instructional programs. Effective and efficient use of computers in education is contingent upon optimizing its effects on learning, while meeting the individualized needs of the student. Only when instructors accept the responsibility for these important tasks, will the educational benefits of CAI be fully realized.

A review of the significant variables associated with this study causes this researcher to comment on several important factors. First, there is consistency among cognitive indicators such as GPI and SAT scores in predicting learning, no matter what type of instructional techniques are utilized. This finding supports the use of these individual variables as predictors of learning. Secondly, the significant findings associated with marital status, increased age, hours of outside employment, negative attitudinal factors, educational and experiential indicators suggests a need to look more closely at these important factors. Looking at the specific data in relation to the developmental profile of the older student, one questions if this profile might be more representative of the non-traditional (older) student than of the average generic student. For example, nontraditional students are often married and older. They are paying for their own education and often need to work longer hours. Their educational and experiential backgrounds are very diverse. While the majority of students in this study were under 25 years and representative of the average student, there were a number of older adults included in this sample that may have contributed to these findings. Thirdly, there was some concern for the significant relationship of negative attitudes toward the lessons and successful learning outcomes. Since the advent of computers as a primary instructional modality was relatively new to this cohort of students, one might question if these findings might change as students have more experiences with computers within the educational domain. This factor needs to be more closely explored as educators continue to develop and design educational programs for computerized instruction.

RECOMMENDATIONS FOR FUTURE RESEARCH

Focus on learner needs provides the basis for planning all educa-

tional interventions. Throughout time, educators have attempted to correlate instructional techniques and the educational objectives of the setting with the developmental needs of the learner. It is now time for educators to go beyond the developmental needs of the learner to the more specific needs of the individual. With the advent of computers in education, the implications of this important aspect of education must be more comprehensively researched in terms of the impact of computers in the learning processes of the individual student.

Based on the findings of this study and other limited, but relevant research, numerous individual traits have been identified to correlate with learning specific content by CAI. Individual learner traits are a prime factor in the learning process and, thus, they are worthy of continued research as educators begin to develop the potential of computers in education. More specifically, this researcher suggests that: (1) Future studies continue to go beyond the findings of simple descriptive statistics and utilize the more sophisticated data produced by inferential statistics to explore the relationship of selected individual traits and learning outcomes; (2) Researchers continue to explore new variables that might contribute to learning with CAI; (3) Researchers replicate this study focusing only on the traditional generic student.

Reference

(1) Boettcher, E., Alderson, S. and Saccucci, M. (1981). A Comparison of Computer-Assisted Instruction Versus Printed Instruction on Student Learning in the Cognitive Categories of Knowledge and Application. Journal of Computer-Based Instruction, 8, 13-17.

COMPUTER PHOBIA. HOW TO HELP PEOPLE OVERCOME IT

JANE TIMPKE, INSTRUCTIONAL DESIGNER
STATEWIDE NURSING PROGRAM
THE CONSORTIUM OF THE CALFORNIA STATE UNIVERSITY
LONG BEACH, CALIFORNIA
ADCIS CONFERENCE 1985

In setting up a system for utilizing computers for instruction in higher education, there have been some challenges. The one that is the most prevalent is "computer phobia." This phobia can be very real. It can be the result of the fear of failing, looking foolish, or of the unknown. Grobes (1984) indicates nurses` fears of computers is derived primarily from a lack of familiarity and understanding about their practical uses in health care settings.

When comments are made such as, "I think all of the faculty should have a computer in their offices, but I don`t need one because I have a secretary," computer phobia may be the hidden reason. If a demonstration of the computer has just been given, and the participant indicates it looks easy but would rather try it when there is more time, computer phobia may be the reason.

Computer phobia is a present-day reality, but it is becoming less of a barrier with the advent of more and more computers usage in educational and administrative tasks. With the increasing utilization of computer orientation programs, computer phobia will fade into the background. Some things that have been helpful in assisting people to overcome this phobia will be presented in this paper. These include: assessing where the individuals are on the "phobia line"; helping the individuals to see that there is no basis for the fear; being enthusiastic; encouraging attendance at workshops; and conducting training sessions.

ASSESS THE PEOPLE INVOLVED

When training people in how to use the computer, one of the things that can be done to eliminate phobia, if it exists, is to take time to assess the individual or group. find out their likes and dislikes, find out what motivates the individuals. Take time to get to know the people before beginning a training program. Gradually ease into finding out how they feel about having the computer in their work or school setting. What experience have these individuals had with a computer? A wide range of experience and ideas will probably be found, from seeing the computer as a life threatening experience to being the best thing that ever happened to modern society. When it is known how people view the computer, it will be easier to know how to proceed with a training program.

HELPING INDIVIDUALS REALIZE THAT THERE IS NO BASIS FOR FEAR

If the fear of being replaced is found, make it clear that the computer is not a threat to anyone`s job. Point out that the computer may bring about a new focus to their job, but it is not a replacement. Helping people see how the computer can save them time and money in some way that is relevant to their situation may be useful. When introducing the computer to the nursing education staff in one hospital, a program that tests the hospital staff in CPR was shown. The nursing administration could see that having a program that could test the staff`s ability and knowledge base on CPR could save time and money. The program, CPR Certification, was developed by T.J. Designs (1984). It is easy to use and requires little training; therefore, it was a good way to show a practical use of the computer.

Students will probably have the least

amount of phobia about computers than other groups, but it is helpful to point out the privacy of learning on a computer. In using a computer program entitled Calculation of Drug Dosages, students indicated they liked the idea of being able to learn how to calculate drag dosages without an instructor looking over their shoulder and discovering a lack in basic math skills (Timpke, 1983). The computer program analyzes the students' skills in basic math and drug calculation. If they are lacking in some areas, they are referred to a workbook to study before taking another exam. In this computer program, the students can retake it until they achieve a desired score. When the student achieves the desired score, the instructor is given a printout of the results. Introducing the computer to students with nonthreatening programs similar to this one assists in eliminating computer phobia.

Another tactic to alleviate phobia is to assure the students that it is not possible to damage or hurt the computer by pressing keys on the keyboard. When someone realizes that it is okay to press any key without hurting the computer, they feel more free to experiment.

According to the audience, using fun games to introduce the computer can be quite effective. Using a game that does not require much thinking can be a nonthreatening experience and, when so used, has been found to have positive results. However, the type of individual or group must be considered. Some individuals would feel that playing games is a waste of time.

BE ENTHUSIASTIC ABOUT WHAT THE COMPUTER CAN DO

Share some of the reasons the computer was selected to be an instructional tool in the work or school setting. These reasons may include saving time and money, orientation to new technology, giving immediate feedback and allowing for individual differences.

Using the computer to teach concepts can save the instructor time. This time can then be used for clarifying concepts. Using a well-designed computer program can save the student time as well. If a student has to read to draw out the main concepts it will take longer than if the information is carefully designed into an interactive computer program. When there is a saving in time, a financial savings usually results.

Orienting the students to new computer technology will make them more valuable in the work force. Most hospitals are relying on some type of computer to assist in many aspects. A student who has had an orientation to computers will probably learn a new computer system more readily than someone who has not had an orientation.

The computer can give immediate feedback as to how the student is progressing. Many times in traditional programs a student has to wait hours, days, or weeks before receiving feedback from an instructor. With each computer lesson or test the student is given immediate feedback and can review or evaluate any mistakes.

The computer also allows for individual differences. If a student requires very little time to grasp a concept, the computer can recognize it and therefore can spend only the amount of time needed. If the student requires more time, the program can be designed to allow as much time as necessary.

There are many things that could be discussed with individuals that are in a computer orientation, depending upon the situation. If the presenter is enthusiastic and relates what the computer can do to things that are applicable to the group, the anxiety level of the group will be lowered.

ENCOURAGE ATTENDANCE AT WORKSHOPS, CONFERENCES, AND COMPUTER CLUBS

From attending workshops, conferences and computer clubs, people begin to feel more comfortable with computers. After assessing the needs of the individual, guidance to workshops on word processing, data base management, hospital systems, software selection, etc., may be helpful. Encouraging participation at computer conferences can also be very beneficial. Finding out about new technology and about what others are doing in the field can give a broad picture of what is happening in the ever changing computer field.

TRAINING

There must be an adequate number of personnel and time devoted to training. Allowing enough time for a training program that includes a demonstration, followed by a hands-on experience, is vital. According to Ronald, (1983), hands-on activities are used to decrease learners' fears and assist them in seeing the computer as a useful tool for personal and professional use.

The Statewide Nursing Program of The Consortium of The Calfornia State University has a nontraditional nursing program for the working RN to earn a BSN

degree. The computer is being used as a tool to teach part of the curriculum. In order to implement the use of the computer, a series of events had to occur. First, Learning Resource Centers had to be established throughout the state that would house the computers and make the computers available at hours convenient to the working student. Second, the instructors and learning resource personnel had to receive a training program. It is too early to report on how the system is working, but one thing was found to be helpful. The students like to work in pairs when learning how to use the computer. This seems to be a more enjoyable experience for them.

Helping people to overcome computer phobia can be challenging, yet rewarding. Some of the things previously stated, such as assessment, enthusiasm, encouragement, and training, have been found to be helpful.

REFERENCES

Grobes, S. (1984, June). Journal of Nursing Education, 23(6).

Ronald, J. S. (1983). Learning needs and attitudes of nursing education with respect to computers. In R. E. Dayoff (Ed.)., Proceeding of the Seventh Annual Symposium on Computer Applications in Medical Care, Los Angeles, CA: Institute of Electrical and Electronic Engineers Society; pp. 523-527.

Timpke, S., & Janney, C. P. (1981, June). Teaching drug dosage by computer. Nursing Outlook, 29, 376-377.

T. J. Designs, 5905 Ironwood, Rancho Palos Verdes, CA., 90274

A Survey Study of Computers in Nursing Education

Barbara S. Thomas, Ph.D. University of Iowa

ABSTRACT

The purpose of this survey was to describe the status of computing in contemporary professional nursing programs, and to identify and rank order barriers which inhibit growth of instructional computing in nursing.

A sample of 157 (51%) usable responses from deans and directors of National League for Nursing (NLN) baccalaureate and higher degree programs was analyzed to describe hardware in use, software/applications in use, faculty development regarding instructional computing, barriers to growth of instructional computing in nursing, and the relative importance of developing instructional computing materials in promotion and salary decisions for nursing faculty.

Faculty use of mainframes is more than graduate students' use, which is more than undergraduate students' use. The situation for microcomputers is the reverse. APPLES outnumber IBM-PCs with TRS-80s in third place. A variety of arrangements with school computing centers and other departments were reported. A number of barriers were reported with faculty interest and development, costs, and lack of useful software mentioned most frequently. CAI program development is viewed dismally low in current promotion and salary decisions for nursing faculty.

This survey study was completed at the University of Iowa with support from the Division of Nursing (Special Projects Grant No. 1 D10 NU270075). The assistance and cooperation of Dr. Donald McClain, Co-Project Director, is gratefully acknowledged.

88

Computers in Nursing Education

Introduction

Although the worth of computer-based instruction in nursing education has been espoused for many years, the actual use of computer technology has remained the exception rather than the rule until recent years. With some important exceptions, the literature from 1970-77 contained little more than statements advocating computer-based instruction, but the picture since about 1978 has changed. More and more applications are being reported, and a few even report evaluative studies of outcomes from computer-assisted instruction (CAI).

In 1983 and 1984, more than ten regional conferences were held on computers in nursing, and nursing membership in international and national computing organizations became much more visible. It is difficult to determine, however, whether the increased visibility is based more on the enthusiasm of a few or an expanded use of computers by many nurse educators. This survey study was launched to describe current use of computers in nursing education accurately and to identify barriers to growth of educational applications.

Review of the Literature

For over a decade, nursing leaders have pointed to computer-based instruction as a means of solving several problems in nursing education. Computer-assisted instruction (CAI) and computer-managed instruction (CMI) were seen as ways of individualizing instruction and allowing nursing students to pace themselves. They could repeat certain drill and practice or tutorial CAI lessons as often as they wished. The computer serves as a tireless, consistent teacher, one which is available many hours each day. Such computer-based learning experiences, when used to replace faculty-students interactions for routine tasks, allowed more time for clinical teaching and problem solving activities. CAI's potential for clinical instruction is also high as de Tornyay noted some years ago:

> Students could initially become involved with a "hypothetical" patient through a computerized program, identify nursing problems, test solutions, and find out the results of their interventions without involving "real patients" with all the inherent dangers requiring close supervision and consequent loss of student experimentation and without the problems involved in the clinical areas that are scarce in the community. (de Tornyay, 1970, p.34-35)

Computer-managed instruction (CMI) has a number of advantages also. Data about students' competencies as they enter the nursing program or individual course, can be collected and summarized easily and quickly. Students' progress in completing various kinds of learning activities can be monitored, with the information also used to prescribe diverse tasks for students with different competencies and learning styles.

Although the many advantages of computer-based instruction have been noted for many years, its growth in nursing education has been slow and faltering. Costs, transportability problems and lack of high quality software have been reported frequently as the primary difficulties. (Please see, for example: Reed, Collart, and Ertel, 1982; Landureth and Lamendola, 1973; Silva, 1973; and Porter, 1978.)

Although microcomputers have reduced the cost and transportability problems, the lack of high quality software is still a limiting factor. The efforts of the most able, experienced teachers are needed, and these individuals have been and remain in short supply. However, the appearance of Software Exchanges and a new journal devoted entirely to computing in nursing provide firm evidence that the software challenge is being met.

The issue of faculty development needs has been a recurrent theme for some years. Armstrong (1983) lists various approaches to faculty development including computer literacy courses, workshops, short courses and one-to-one technical assistance. She describes practical approaches for giving faculty and students sufficient exposure and hands-on experience. Armstrong's suggestion that faculty should be encouraged to take home a microcomputer for a weekend is excellent. I also concur in noting that word-processing software is a good introduction to computers. For several years, I have introduced students to the use of terminals for data analysis by means of text management problems on WYLBUR. Text seems to be a less threatening way to begin than number crunching. The primary goal should be to furnish nurses with on-going, convenient opportunities to learn about computing. The one-shot effort is simply not adequate.

Another barrier to growth of CAI/CMI was identified in a survey study of health sciences faculty who were also members of the Association for the

Development of Computer-based Instructional Systems (ADCIS). Data gathered from faculty in dentistry, medicine, nursing and pharmacy were very consistent in affirming that "publish or perish" flourished in health professions education. Since work on computer-based instructional materials did not contribute significantly to positive promotion or salary decisions, many faculty reported that they moved back into more traditional, scholarly pursuits after a brief period of developing CAI materials (Thomas, 1975).

Since no current information was found about either the growth or the relative importance of the problems associated with implementing computer-based instruction in nursing, a descriptive study was initiated to do just that.

Statement of the Problem

The purpose of this study was to describe the status of computing in nursing education and to identify factors which inhibit its growth. Specifically, the following research questions were posed:

1. How much are different kinds of computers used in nursing programs?
   a. What are the relative use levels of different kinds of hardware for faculty, graduate students and undergraduate students?
   b. How many and what kind of microcomputers are being used in nursing programs currently?
   c. What arrangements have deans and directors of nursing programs made with other units or departments for use of hardware?
2. What kinds of software/applications are being used by faculty and students in professional nursing programs?
3. What is the status of faculty development in instructional computing?
4. What are the most important barriers to the growth of instructional computing in nursing education?
5. What is the relative importance of development of CAI/CMI materials in promotion and salary decisions?

In this study, the three levels of hardware were mainframe or maxi, mini and microcomputers. "Super-minis" were not included. The study was limited to instructional uses of computers in nursing.

Methodology

Population and Sample

The study population was defined as accredited programs of professional nursing. The sampling frame was the list of National League for Nursing (NLN) accredited baccalaureate and higher degree programs, n = 305. The sample was the 157 usable survey instruments which were returned, making the response rate 51%. Questionnaires were returned from large and small schools, from both public and private schools, and from every geographic location. Comparison of the sample statistics with NLN data revealed that the size and public/private categories wee approximately proportional to those in the population. Slightly more responses came from western programs than a proportionate sample would provide. It should be noted as a limitation that the results summarized in this report represent slightly more than half the schools in the sampling frame.

Data Collection and Analysis

A mailed questionnaire was used to collect data from deans and directors of accredited professional nursing programs in November 1983. The questionnaire which consisted of five sections was designed for short answer, computerized analyses. Frequencies, per cents, means, standards deviations and rankings were computed.

Results and Discussion

Hardware

The deans and directors' report of their faculty's and students' use of mainframe, minicomputers and microcomputers revealed that faculty use of mainframes and minicomputers is greater than graduate students' use, which is greater than undergraduate students' use. Faculty use of microcomputers was reported at about the same level as for mainframes. However, more undergraduate than graduate students use microcomputers. Only 20 percent reported use of minicomputers; it is possible that mainframes were over-reported and minicomputers under-reported because of a fairly common misconception that any time-sharing system is a mainframe-based network.

Figure 1 summarizes the data regarding numbers of different kinds of microcomputers in nursing programs. Approximately one-third of the program directors reported ownership of 148 APPLE systems. The mode was one while the maximum was fifteen. The number of

APPLES is almost twice that of the IBM-PCs and more than four times the TRS-80s. Some of the deans and directors reported having several different kinds, thus the system totals are used in Figure 1 to indicate how many microcomputers are in place in nursing programs. The number of item omissions was interpreted as programs with no microcomputers to report, the smallest (No Response) NR value, 103, suggests that about two-thirds of the nursing programs have no microcomputers.

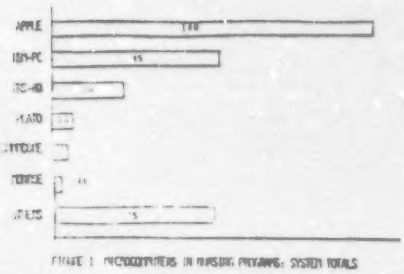


FIGURE 1 MICROCOMPUTERS IN NURSING PROGRAMS: SYSTEM TOTALS

Concerning arrangements for use of microcomputers in other units, the most frequently reported alternatives were use of hardware in other departments, the computing center or lab, microcomputers in faculty offices, the library, unspecified access, and terminals in dorm rooms in that order. Included among the other departments were computer science, chemistry, nutrition and health science centers. Four deans and directors reported that personal purchase of microcomputers was the only access their faculty and students had.

In response to the question on availability of peripherals for microcomputers, 47 (30%) programs reported videodisc/videotape; 31 (20%) graphics tablets; 40 (25%) hard discs; 18 (11%) audio discs; 16 (10%) headphones; 25 (16%) plotters; 96 (61%) printers; 5 (3%) speech synthesizers; 6 (4%) random access slide projectors; and 4 (2%) "other". It is likely that the 47 videodisc/videotapes are largely videotape machines. Although the potential of videodisc technology is remarkable, costs for development and equipment seriously inhibit its use at this time. The large number of printers is consistent with the reported heavy use of computers for word processing and data analysis.

Place of Computing in Nursing Education

Forty (25%) respondents report having made explicit provisions for computer literacy in their programs. Fourteen (9%) offer a computer literacy course as a nursing elective, while eight (5%) each reported required courses in nursing or other departments. Five (3%) reported that electives in other departments meet their computer literacy needs, and one (.6%) each reported use of computers in research and in health care agencies as their only provisions for computer literacy. Nineteen (12%) reported that computer literacy content is still in the planning stages at their institutions. Others note that nursing students gain information about computers in their pre-nursing course-work.

Figure 2 summarizes the data reporting faculty, staff and student use of computers in various applications. Word processing is most common, followed by use of statistical packages, computerized information retrieval, CAI and CMI, and retrieval of patient data in that order. Word prccessing was most used by staff, while statistical packages were reported most frequently for faculty and graduate students. Undergraduate students use computers most frequently for instruction and retrieval of patient data.

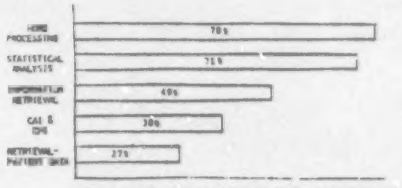


FIGURE 2 USE OF COMPUTERS BY FACULTY, STAFF, AND STUDENTS

One-hundred-two (102, 65%) deans and directors reported having made special provisions for faculty development concerning computing. Some respondents reported several kinds of programs. Seventy-three (46%) indicated instructional applications were the focus; 68 (43%) stated that statistical analysis of research data was the focus; 39 (25%) reported information retrieval; and 56 (36%) identified word processing, with the qualifier that it was primarily for staff rather than faculty.

Respondents were asked to estimate the percentage of their faculty at various skill levels. These data showed that faculty's lack of interest or skills was perceived by the deans and directors to be a serious block to computing in nursing education. Only 44 (28%) deans and directors reported having any faculty with expert computing skills; the range was from 1-30% of the faculty in these programs. The mean percentage of faculty with expert skills in these 44 programs was 5.50%. "Moderately skilled" faculty were reported by 102 (65%) deans and directors. The range was enormous, from 1-99%, and the mean was 13.44. Ninety per cent (n = 142) of the deans and directors reported that anywhere from 1-93% of their faculties were at the "novice" skill level. The mean was 29.48. "Unskilled" faculty were reported by 121 (77%) deans and directors to range from 5-99% of the faculties. The mean was a grim 62.59%. The standard deviations indicate extreme variability in the estimates of faculty skill levels, as do the ranges.

Barriers Inhibiting Growth of Instructional Computing

Several factors have been identified as inhibiting growth in instructional computing. The two lower and two upper categories were combined, and a median was computed for each factor in order to clarify findings. Of these factors, faculty's lack of opportunities to learn, minimal skills, costs, and the paucity of useful software were identified as the most important factors. Other items were identified as lack of information about software and hardware, faculty's lack of interest, lack of evidence of worth and transportability problems.

Sixty-five respondents (41% of the sample) identified additional deterrents without rating the importance of the items. Unspecified financial concerns, faculty's lack of time or support for computing, building renovation needs, lack of technical assistance and the fact that rewards in higher education focus on other activities were mentioned. Six respondents reported that computing was not compatible with or relevant to their curricula.

Certainly, growth of instructional computing in nursing curricula has been slower than predicted in the early seventies. Of the 157 respondents, 118 (75%) reported that they had neither developed nor purchased any software for instructional applications. Only 43 (27%) reported that past or present faculty in their programs had developed computer-based instructional materials to augment their teaching.

The notion that the reward system in higher education focuses on research and publication rather than on development of instructional materials was mentioned by only three respondents. This was surprising in view of the number of times that faculty have reported informally that they could not afford to work on computer-based instructional materials because of the pressures to do research and publish. Possibly, the importance of this factor is under-represented because this survey reports deans' and directors' perceptions, not those of nursing faculty. In addition, it had to be identified in response to an open-ended question.

Deans and directors were asked to rank (1 = least important, to 3 = most important) the relative importance of teaching, research and publication, and service in promotion-tenure decisions. Clearly, excellence in teaching was perceived to be the most important consideration; it was ranked first by 108 (69%) respondents. However, 52 respondents indicated that the publication of research in refereed journals is an absolute requirement for the award of tenure. Among those who indicated that it is not, various sorts of accomplishments were listed as serving as substitutes, e.g.: supervising doctoral studies, other publications, outstanding service, and activities in professional organizations. Development of computer-based instructional materials was not mentioned, although one might expect the heavy emphasis on excellence in teaching to be coupled with rewards for originality and productivity in relation to developing instructional materials.

To determine the importance of CAI development as evidence of scholarship, respondents were asked to indicate the relative weight given to selected accomplishments in making promotion, tenure or salary decisions. The most important indicators of scholarly productivity were identified by combining the n's for "Moderately Important" and "Very Important". The most important indicators were: research article in a refereed journal; textbooks for graduate and undergraduate students; and nonresearch article in a refereed journal. CAI program development was ranked next to last of the thirteen items, with 22 respondents indicating that it is not considered at all. No doubt the fact that few faculty have developed CAI programs accounts for its low ratings.

Salary considerations were also ranked by combining the ratings of "Moderately Important" and "Very Important" and "Not Important" and Slightly Important". The results indicate that the most important factors in making salary decisions to be students' evaluation of teaching, research publications, administrators' and peers' evaluation of teaching, publications about issues, external funding and service activities. CAI development was rated last of 12 factors. These data provide firm if indirect support for the notion that the current reward system in nursing education is a serious deterrent to growth of computing in nursing.

Implications

Although the nonresponse rate (49%) should be noted as a limitation of this study, the responses did come from all parts of the country and all kinds of nursing programs. It is hoped that survey findings provide a reasonably accurate picture of computing in contemporary nursing education. It should be remembered that these findings represent the perceptions of deans and directors. In some areas such as opinions about skill levels or deterrents to growth of computing, faculty perceptions might paint a different picture.

Clearly the potential of CAI for augmenting instruction in nursing education is not being realized at the present time. To improve this picture, both provisions and incentives are needed for faculty development. One indicator of favorable trends is the fact that continuing education programs on computing for nurse educators have proliferated recently. More than a dozen brochures describing such conferences have been distributed during the past year. It will be interesting to reassess faculty skill levels and use of computers in the next few years. As the cadre of skilled faculty members increases, the development of useful software should increase also. Finding the extended time needed to develop CAI programs, however, remains a serious problem. Faculty cannot afford to devote time and energy to CAI development projects when they are pressed to teach long hours, earn advanced degrees in their "spare time", complete and publish research, provide service to community and professional organizations and serve on committees. If nursing leaders are serious about their interest in CAI, it is time to reassess faculty workloads as well as the current reward system.

References

Armstrong, Myrna, "Paving the Way for More Effective Computer Usage", Nurs Health Care, 1983, Dec. 4(10), 557-9.

DeTornyay, R., "Instructional Technology and Nursing Education," Journal of Nursing Education, 9:3-8, 34-35, 1970.

Landureth, J.L. and Lamendola, J.A., "Computers and Nursing Education", Hospitals, 47:99-102, March 1, 1973.

Porter, S.F., "Application of Computer-Assisted Instruction to Continuing Education in Nursing", J Cont Ed Nursing, 9:5-9; 1978.

Reed, F.D., Collart, M.E., and Ertel, P.Y., "Computer Assisted Instruction for Continued Learning", Am J Nurs, 72:2035-2039, November 1972.

Silva, M.C., "Nursing Education in the Computer Age", Nurs Outlook, 21:94-97, February 1973.

Thomas, B., "Evaluation of CAI Activities in Promotion-Tenure Decisions", Unpublished Report, 1975.

Promoting C[illegible]g in Nursing Education:

A Delphi Study

Barbara S. Thomas, Ph.D. University of Iowa

ABSTRACT

The purpose of this study was to clarify and amplify 18 barriers to instructional computing which were reported in a survey study of computers in nursing education. A stratified, random sample of 32 deans and directors participated in the study.

The two stage Delphi study produced rankings of barriers in terms of their importance and the feasibility of changing them. The primary topics were costs of hardware and software, faculty development concerns, lack of funds and personnel for technical assistance, and lack of space for facilities and in curricula. Practical solutions were identified and the group set priorities among their remedies.

*This work was completed with support from the Divison of Nursing (DHHS) Special Projects Grant No. 1D10 NU 27C076. The assistance and cooperation of Dr. Donald McClain, Co-Project Director is gratefully acknowledged.

Delphi Study

Introduction

Instructional computing has grown much more slowly in nursing education than was expected when the usefulness of modes such as drill and practice and simulations were described (Bitzer, 1970; deTornyay, 1970; Valish, 1975 and Boyd, 1975). Although some nursing programs rely heavily on computers for instruction as well as for practice and research applications, it is more common to find faculty beginning just now to plan for integrating computers into their programs. In our recent survey study of computing in nursing, 40 programs reported having made provisions for computer literacy, while 19 stated that planning is underway, and 98 indicated neither provisions nor plans.

In the past decade, a number of reasons has been advanced for the slow growth of computer-based instruction. Costs and transportability/ compatibility difficulties were the primary problems when mainframe and minicomputers were the only available hardware. (Please see for example, Bitzer and Bitzer, 1973; Braunfield, 1964; and Levine and Wiener, 1975.)

Although the introduction of microcomputers ameliorated these problems, it has been apparent that the progress envisioned by deTornyay (1970), Collart (1973) and others has not been attained. Cost is still a factor, but faculty's lack of interest and skills and the paucity of useful courseware were also identified by deans and directors as important barriers to instructional computing in professional nursing programs (Thomas and McClain, 1984). The list of 18 barriers identified by deans and directors constituted the list of items for the first stage of the Delphi Study. To clarify and rank these barriers a two-stage study was initiated in November, 1983.

Methodology

Population and Sampling

The Delphi participants were selected from deans and directors who indicated in the earlier Survey of Computing in Nursing Education (1984) that they would be willing to participate. A stratified random sample was drawn with strata based on both geographic region and size. Respondents represented publicly and privately supported schools and also institutions with and without graduate programs. This report is based on responses from 32 deans and directors with representation from both large and small programs in all parts of the country.

Procedures

In the first stage, participants were given survey findings regarding perceived barriers to growth of computing in nursing education. Four categories were identified: 1) Costs, 2) Faculty Development Concerns, 3) Lack of Information, and 4) Status of Computing. Participants were asked to add to the list or revise it, and also to rate each item as to its importance and its feasibility of being changed.

Results

Table 1 summarizes these barriers and the mean ratings, showing that costs of hardware and software were deemed the most important barriers, followed by lack of useful software and faculty's lack of skills or opportunities to learn. With the exception of "Status of Computing: Too new", all mean scores were above the slight importance rating level. Seven of the items' importance were rated between slight and moderate importance.

The feasibility ratings ranged from 1.23 to 2.47 on a scale of 0-3 representing "no feasibility" to "great feasibility". Understandably, The 32 respondents felt that the lack of information about hardware and software, along with faculty's lack of skills or opportunities to learn were three barriers most amenable to change. Understandably, the factors least feasible for change were rapid changes in computing, building renovations, lack of space and transportability problems.

Participants were also asked to identify possible remedies for the barriers they considered most important. One or more such factors were described for 16 of the 18 barriers. No ways to ameliorate the following were given: (1) computing in nursing too new, and (2) changes in computing too rapid. For the remaining 16 items, the potential remedies were summarized, categorized and tabulated for the second stage of the Delphi Study. In this second phase, participants were asked to rate the feasibility of each potential remedy on a scale of 0 (no feasibility) to 3 (great feasibility). They were asked also to select and rank the importance of the top five items. These were reversed for computing means; i.e., an item ranked number one was given a weight of five points, a ranking of two was given four points, and so forth. Items left blank were given zero in calculating the means for each barrier.

Table 1
Barriers to Computing in Nursing Education
(n = 32)
no importance - 0 - no feasibility
slight importance - 1 - slight feasibility
moderate importance -2- moderate feasibility
great importance - 3 - great feasibility

Barrier	Importance $\bar{X}$	Feasibility for Change $\bar{X}$
Costs		
1. Hardware	2.44	1.72
2. Software	2.44	1.84
3. Lack of space	1.53	1.68
4. Building renovations needed	1.37	1.33
Faculty Development Concerns		
1. Lack of interest	1.66	2.29
2. Lack of skills or opportunities to learn	2.16	2.37
3. Lack of technical assistance	2.06	1.87
4. Lack of incentives, rewards	1.75	1.81
5. Lack of "space" in curriculum	1.66	2.12
Lack of Information		
1. Hardware	1.87	2.42
2. Software	2.07	2.43
3. Evidence of worth	2.10	2.28
Status of Computing		
1. Too new	0.84	1.97
2. Changes too rapid	1.61	1.23
3. Lack of useful software	2.35	1.90
4. Transportability/compatibility problems	1.77	1.63
5. Inconsistent with nursing program focus	1.06	1.93
6. Policy - support for central computing facility		

The importance ratings for the 18 barriers were as follows:

	Mean Importance
1. Cost of hardware excessive for instructional applications with large student bodies	3.16
2. Cost of software	2.38
6. Faculty's lack of skills or opportunities to learn	1.83
5. Lack of faculty interest in computer applications	1.53
4. Costs of training faculty and staff	1.13
7. Lack of technical assistance for faculty	1.07
9. Lack of "space" in curriculum	0.87
3. Lack of space; costs of building renovations	0.87
15. Lack of useful software	0.83
12. Lack of evidence of worth; i.e., that CAI is cost-effective	0.77
14. Changes in computing too rapid	0.50
8. Lack of incentives, rewards	0.50
13. Computing in nursing too new	0.43
11. Lack of information about software	0.43
16. Transportability/compatibility problems	0.33
18. Policy of supporting a central computing facility inhibits ready access for nursing faculty and students	0.33
17. Inconsistent with nursing program focus	0.17
10. Lack of information about hardware	0.13

Costs were identified clearly as the most serious deterrent to growth of computing in nursing education. Various aspects of faculty development, interests and commitments were judged as very important also, as well as building renovation costs and needed curriculum modifications. These items were followed by: "lack of useful software" and "lack of evidence of worth, i.e., that CAI is cost-effective". "Changes in computing too rapid" and "lack of incentives, rewards" were ranked equally. Four specified "computing in nursing too new" which ranked thirteenth, along with "lack of information about software". "Transportability/compatibility problems", ranked fifteenth. These data verify the rapid changes in transportability/ compatibility problems with the advent of microcomputer technology. Also ranked fifteenth was "policy of supporting a central computing facility inhibits ready access for nursing faculty and students." The last factors in the list were the items "inconsistent with nursing program focus" and "lack of information about hardware." Opinions of deans and directors were diverse, with only two items reported in the top five for more than half the sample. Each of the 18 items was ranked among the five most important barriers by at least one participant.

The mean feasibility ratings given to the two most important items: costs of hardware and software were as follows:

	Mean Feasibility
Costs of Hardware:	
External funding, grants	1.70
Interdepartmental or intercollegiate cooperation	1.70
Solicitation of industrial support	1.17
Changed focus to nursing practice applications	1.13
Solicitation of community support	1.07
Rechanneled general funds	0.97
Cost of Software:	
Increased numbers will generate competition, bringing costs down	2.07
Increased budget line for useful software	1.47
Increased use of courseware authoring systems should reduce costs	1.40
Dissemination of information on marketing to stimulate faculty development of software	1.30
Encouragement of faculty to develop software (with small payments)	1.20
Provision of some faculty released time	1.20
Hiring of a programmer to work with faculty	1.10

Thirty respondents provided usable data for these comparisons. The 0-3 weights were simply averaged. External funding as well as cooperation among colleges and departments were seen as the most feasible ways of dealing with hardware costs. Changing the focus to nursing practice applications was identified as feasible also. The fact that clinical applications and nursing service administration topics are being addressed enthusiastically at current nursing computer conferences support this factor as a significant trend. At the same time, it is possible that use of computers which are available in many health care settings might provide nurse educators with scarce resources for instructional applications as well as patient care or administrative applications. Finding funds in general budgets was the least feasible alternative to hardware costs.

The most feasible ways of dealing with software costs focused on the beliefs that competition will bring costs down, increased budget lines will enable more purchases of useful software, and that increased use of courseware authoring systems will reduce costs. Some support was given also to giving faculty small payments or released time for software development and to hiring a programmer to assist faculty.

The data regarding faculty concerns shows the remedies for faculty development concerns to be as follows:

	Mean Feasibility
Faculty's Lack of Skills or Opportunities to Learn:	
Develop educational programs for faculty	2.37
Promote team approach - faculty with computer specialists	1.77
Provide time for continuing practice as well as initial study	1.63
Provide hardware/software and time for faculty to play with it	1.50
Budget funds for inservice education	1.43
Provide released time for faculty to attend programs of their choice (external or internal)	1.30
Lack of Faculty Interest in Computer Applications:	
Conduct seminars/CE workshops	2.33
Establish a task force to identify needs, resources and a plan	2.13
Demonstrate the integration of computers with practice functions of service, education and research	2.13
Develop positive computer environment by establishing new leaders with computer competence among faculty	2.03
Provide incentives for becoming computer literate	1.80
Stimulation of faculty interest by students	1.50
Lack of Technical Assistance of Faculty:	
Promote faculty expertise (as above)	1.80
Provide personnel to work with faculty	1.40
Hire a programmer	0.70
Lack of "Space" in Curriculum (Faculty Attitudes):	
Faculty continuing education	1.80
Faculty development concerning integration of computers with curriculum content	1.77
Evaluation of curriculum design	1.60
Creative planning for use of faculty and student time	1.37
Lack of time for faculty development of software	1.07
Pressure to focus on state boards: should recognize computing as a means to this end	0.97
Released time for interested faculty	0.93
Cost of Training Faculty and Staff:	
Allocation of internal funds for faculty and staff workshops, institutes or classes	1.97
Adopt computer literacy programs for faculty	1.90
Reach consensus concerning skill level needed by faculty and provide for it from budgeted funds	1.47
Send interested faculty to visit developed programs	1.43
Released time for interested faculty to assist others	1.10

Practical, realistic remedies for faculty development included inservice education, initiation of a team approach and giving faculty released time to learn and maintain skills. Some felt that students' interest will stimulate faculty interest. Provision of technical assistance was rated rather low.

Considerable attention was given to the problem of "lack of space" in the curriculum. Several related remedies suggest that if faculty become more informed, they will see ways of integrating computing with other components of the nursing curriculum and for assisting students to gain higher levels of clinical skills as well as more nursing knowledge.

Ways for improving faculty skills were largely internal measures although sending faculty to visit well-developed programs was listed. More highly rated, however, were internally funded and

organized workshops, and computer literacy programs. It was noted that faculty consensus about skill levels needed was rated between slight and moderate feasibility.

The responses related to lack of resources, incentives and transportability were as follows:

	Mean Feasibility
Lack of Space; Costs of Building Renovations:	
General center for all departments	1.77
Grant support	1.30
Budget changes	1.03
Lack of Useful Software:	
Encourage in-house development of software by faculty	1.73
Provide technical assistance to interested faculty	1.67
Lack of Evidence of Worth:	
Insist on opportunities to preview software	2.27
Research	2.03
Lack of Incentives, Rewards:	
Develop incentives for faculty authorship	1.67
Develop incentives for faculty study	1.63
Change Promotion/Tenure policies so that software production is as acceptable as an article in a refereed journal	1.47
Reward faculty who integrate new technology into their practice, teaching and research	1.43
Provide released time	1.17
Tranportability/Compatibility Problems:	
Promote faculty development of programs for hardware in place	1.70
Quad links might resolve problems in some systems	1.03

Feasibility rating ranged from 1.03-2.33. The two factors with ratings approximately 2.00 concerned lack of evidence of worth. They were insistence on opportunities to preview software and research. The lowest ratings were related to lack of space and costs of building renovatings, provision of released time and

The data concerning lack of information about hardware and software, nursing program focus and lack of ready access to computers in nursing programs resulted in the following list of factors and their mean feasibilities:

The feasibility ratings were quite homogeneous, ranging from 1.77 to 2.33. The most feasible factors were informing software suppliers what's needed for nursing programs, improving communication generally, and obtaining information about ongoing costs. Less feasible were seminars, stimulating program development, dissemination of informtion via in-house publications or subscribing to computing magazines.

Having a general computing center was noted as a feasible solution to cost-space problems. However, the use of a center was viewed as an inhibiting factor. Probably, nursing faculty and students are less comfortable in such a center when they are only beginning to acquire computing skills. In-house availability of equipment was preferred for this reason as well as for physical accessibility.

In-house development of software can be stimulated by providing inservice education incentives and technical support. Using a team approach where the nurse educator functions as a content expert and works with instructional designers and programmers was viewed as an effective way to deal with time-cost barriers.

Previewing software prior to purchase and evaluation were noted as both important and feasible for dealing with the problem, lack of evidence of worth. The need for authors and publishers to gather sound data about the effectiveness of CAI materials is clear. Some of these items overlap with factors identified in terms of cost and faculty concerns. In this study, no great effort was placed on developing mutually exclusive categories because the purpose of the study was to identify as many deterrents and remedies as possible. Some readers may wish to

Barriers and Remedies	Mean Feasibility
Central Computing Facility Inhibits Ready Access	
Work with administrators to determine needs and compatibility issues	2.13
Lack of Information about Software	
Make presentations at faculty meetings	2.03
Obtain dependable information from suppliers	2.03
Create a center at which one could preview software locally from suppliers	2.03
Hold seminars	1.87
Stimulate more variety in software so that programs will "fit" nursing curriculum focus	1.87
Disseminate information via in-house publications	1.80
Subscribe to computing magazines	1.77
Inconsistent with Nursing Program Focus	
Inform suppliers what is needed	2.33
Determine needs precisely and search for appropriate "fit"	2.03
Stimulate more nursing software	1.93
Support in-house development of software	1.93
Lack of Information about Hardware	
Improve communication	2.30
Obtain information about ongoing costs: Supplies and maintenance of equipment	2.17

consider this to be a limitation of the study but should note that nuances in specifying certain topics were preserved by the approach which was taken.

The assumption was also made that stimulating instructional computer applications in nursing is both desirable and important. Twenty-eight of the responding deans and directors agreed with this assumption; one favored growth in computing as of primary importance. Another considered computing as not a primary issue in nursing.

Even though the study was conducted in a two-month period bordering the holidays, there was fine participation from busy deans and directors. Their responses reveal not only a high interest in computer technology, but also a firm grasp of the practical issues.

The task of integrating computer technology into nursing curricula is similar in several ways to the movement of nursing toward a research-based science some years ago. Progress in both has resulted from enormous efforts by a minority of nurse educators with unusually high levels of competence, commitments, and interests. Similarly, resistance to both has been based on misconceptions that research and computer technology are topics rather than tools. Fears that computing in nursing will strain overcrowded nursing curricula must be remedied with effective faculty development programs so that computing is viewed as essential and as a means to an end.

Respondents in this study, by using the Delphi technique to pool their ideas and judgments have provided clear guidelines for needed change. Geographically separated individuals with special knowledge were able to pool their ideas in a simple, straight-forward way. Inhibiting factors were identified and priorities among potential remedies were set. It is hoped that these results will promote effective measures in nursing education.

References

Bitzer, M., "Nursing in the Decade Ahead," AJN, 70:2117, 2118, 1970.

Bitzer, M. and D. Bitzer, "Teaching Nursing by Computer: An Evaluation Study," Comp. and Biol. Med., 3:187-204, 1973.

Braunfield, P. G., "Problems and Prospects of Teaching with a Computer," J Educ Psych, 55:201-211, 1964.

Collart, M. E., "Computer-Assisted Instruction and the Teaching Learning Process," Nurs Out, 21:527-532, 1973.

deTornyay, R., "Instructional Technology and Nursing Education," J. Nurs Ed, 9:3-8, 34, 35, 1970.

Levine, Daniel and Earl Wiener, "Let the Computer Teach It," AJN, 1300-1302, August, 1975.

Thomas, B. and D. McClain, "Microcomputers in Continuing Nursing Education," paper presentation at the Conference, "Nursing Research: New Directions for Practice and Education," University of Rochester School of Nursing; Rochester, New York, 1984.

Valish, A. and N. J. Boyd, "The Role of Computer-Assisted Instruction in Continuing Education of Registered Nurses: An Experimental Study," J Cont Ed Nurs, 6:13-31, 1975.

A Microcomputer Authoring System

for

Clinical Simulations

Richard S. Trynda, D.V.M., M.Ed.

University of Colorado Health Sciences Center
Office of Educational Services
Denver, Colorado

ABSTRACT

This paper describes a microcomputer authoring system which allows for the development of clinical simulations. Knowledge of computer programming is not required. Simulations are created by entering content, logic and a scoring system into a knowledge-data base using a menu-driven editor. A driver program generates the simulation from this knowledge-data base. A printout of user performance data is available using an instructor disk.

INTRODUCTION

The development of computer-based instructional (CBI) software is very labor intensive. Generally, CBI development time increases with the complexity of the software. Estimates to produce one hour of computer-based instruction range from 100 to 300 hours depending upon software design. The need for authoring systems that can decrease this level of effort is clearly demonstrated.

Authoring systems currently exist which allow for the development of frame-oriented CBI software. These systems simplify the development of tutorial, and drill and practice CBI software. However, development of highly interactive simulations using these authoring systems is quite cumbersome. Therefore, a need exists for authoring systems which will decrease the development time for simulations.

The authoring system described in this paper allows for the development of clinical simulations. Three needs motivated the development of this system.

1. To provide experience in differential diagnosis.

Many diseases and conditions initially manifest themselves by presenting identical signs and symptoms. The skilled diagnostician selectively gathers data on a patient to discriminate among tentative diagnoses. Negative evidence will rule out a tentative diagnosis. Positive evidence will assist in confirming a definitive diagnosis. (Clancey & Shortliffe, 1984; Elstein, Shulman & Sprafka, 1978).

2. To provide a constant selection of clinical cases for the student in a health professions' curriculum.

In teaching hospitals and clinics, patients are admitted and discharged on a daily basis. Clinical rotation does not guarantee all students in a class, nor students from one year to the next, will have the same clinical experiences (Jones & Keith, 1983; Trynda, 1979).

3. To assess an individual's problem solving skills.

This is an ongoing concern for health educators. Various strategies, formats and media have been used to assess problem solving skills. Some examples are: low fidelity simulations such as paper format patient management problems, high fidelity simulations using actors to portray patients, and

mini/mainframe computer-based clinical simulations (Elstein, Shulman & Sprafka, 1978; Jones & Keith, 1983; McGuire, Solomon & Bashook, 1976).

The components of this authoring system include an editor, a driver, a data disk and an instructor disk.

FUNCTIONAL DESCRIPTION

The Simulation

In a typical simulation developed with this authoring system, the opening scene presents a patient displaying certain signs and symptoms. Information on the patient is obtained by selecting items from sections labeled history, physical examination, laboratory, radiology, etc. Each item is weighted as to its degree of being helpful or harmful.

Feedback is provided on the order these items should be selected. Feedback is also given regarding which items are required to rule out or confirm a diagnosis.

Once a diagnosis is confirmed, the user is requested to enter their initial management or treatment of the case. A case summary is provided followed by an opportunity to comment on the simulation. Information on the user's performance, their initial management or treatment of the case and their comments are stored at the completion of the simulation for later retrieval and review.

The Editor

To develop a simulation, three types of information are required. These are content, logic and a scoring system.

The content is the text used to create the opening scene, the names for sections and items, the responses, the feedback, the diagnoses and the case summary.

The logic is heuristic in nature. It is used to designate which items must be investigated before, or are prerequisite to, other items. The logic is also used to designate which items are required to rule out or confirm a diagnosis. The logic information is required if the simulation is to be instructional. If the logic is absent, the need for corrective feedback is not detected and the simulation becomes an assessment tool.

The scoring system is used to weight each item as to its degree of being helpful or harmful for a particular clinical simulation. A numerical scale is used. Helpful items are weighted positively, harmful items are weighted negatively and neutral items are assigned a weight of zero (McGuire, Solomon & Bashook, 1976).

The Instructor Disk

The instructor disk allows printouts of user performance data to be generated. This printout includes summaries (by weight and section) of which items have been choosen, the sequence of which items and diagnoses have been choosen, calculated performance scores, the user's initial management or treatment of the case and their comments.

HARDWARE REQUIREMENTS

This authoring system is written in Pascal. Currently, it runs on a 64K Apple IIe with an 80 column card and two disk drives. A printer is recommended for authoring and instructor use. Implementations on other microcomputers are planned. Future versions will incorporate control of a laser videodisc player.

SUMMARY

Simulations can be developed for medicine, nursing, dentistry and veterinary medicine using this authoring system.

While certainly not a substitute for actual patients, a library of computer-based clinical simulations can provide opportunities for exercising cognitive diagnostic skills.

REFERENCES

Clancey, W.J., Shortliffe, E.H. 1984. Readings in Medical Artificial Intelligence: The First Decade . Menlo Park, CA: Addison-Wesley Publishing Company.

Elstein, A.S., Shulman, L.S., and Sprafka, S.A. 1978. Medical Problem Solving: An Analysis of Clinical Reasoning . Cambridge, MA: Harvard University Press.

Jones, G.L. and Keith, K.D. "Computer Clinical Simulations in Health Sciences." The Journal of Computer-Based Instruction , 1983, 9(3), 108-114.

McGuire, C.H., Solomon, L.M., and Bashook, P.G. 1976. Construction and Use of Written Simulations . The Psychological Corporation.

Trynda, R.S. "The Role of Computer-assisted Instruction in a Veterinary Medical Curriculum: An Overview." The Journal of Veterinary Medical Education . 1979, 6(2), 113-116.

MICROCOMPUTERS IN HOME ECONOMICS CURRICULUM

IN CALIFORNIA COMMUNITY COLLEGES

Dr. Marilyn D. Ambrose, Professor and Chair

California State University, Chico

ABSTRACT

This project funded during 1983-84 for approximately $20,000 by the Chancellor's Office of California Community Colleges proposed to:

1. Identify statewide and local needs for microcomputer software in the curriculum areas of foods and nutrition, child development, family life and parenting, textiles and clothing, interior design, personal and family economics, and family resource management;
2. Develop three new software programs;
3. Update the extant annotated bibliography into a resource guide; and
4. Disseminate project materials and activities.

To achieve these objectives, a statewide survey was conducted to identify software needs in the home economics curriculum areas and newly available software. Second, three new software programs were developed by cooperative efforts of consultants, curriculum specialists, home economics instructors, and computer programmers. The current Resource Guide was stored on microcomputer and updated with survey results and the three new software programs.

Dissemination of project materials and activities occurred through a presentation at subject area conferences, by distribution of a brochure about the three new software programs and by distribution of hard copies of the Resource Guide to all community colleges with identified home economics programs, the Chancellor's Office, and ERIC.

RATIONALE

Home economists can and should be in the forefront in this rapidly changing and technological world because of their primary mission of improving the quality of life for individuals and families. One aspect in the complex issues facing us today is the use of computers in the arenas of home, classroom, business, and industry. Reliable trend and forecast data reinforce the necessity of educators being able to understand, utilize, interpret, synthesize, and evaluate the world of computers. For home economists to remain on the cutting edge: (1) technophobia must be overcome; (2) computer literacy in its broadest definition must be enhanced; and (3) innovative implementation of microcomputer software must become an integral component of home economics (H E) programs. To do less, is to deprive students in each arena and level of H E programs the chance to equitably compete and succeed in their chosen professional field. Microcomputers have the greatest potential for teaching and learning, but developing innovative educational materials is lacking in all fields.

Development of appropriate and quality software is a very time-consuming and labor intensive process, requiring skills of specialists and cooperative team efforts of: (1) home economists--to provide ideas and subject matter expertise; (2) computer scientists--to provide programming skills and hardware/software compatability; and (3) educators--to emphasize educational principles in developing quality programs for a variety of learner levels.

Of these three, there is a lack of home economics personnel familiar with computers, programming, and applications.

MICROCOMPUTERS IN HOME ECONOMICS CURRICULUM

An initial grant provided funds to conduct workshops throughout California to enhance computer literacy of home economists. This project continued that focus through development of additional software and updating of a microcomputer Resource Guide.

OBJECTIVES, PROCEDURES, ACTIVITIES

The goal of this project was to improve the availability of computer software curriculum for H E programs in the California Community Colleges (CCC). The objectives were:

(1) To survey H E programs at all CCC, vendors and computer users for what software needs exist in foods and nutrition, child development, family life and parenting, textiles and clothing, interior design, personal and family economics, and family resource management; what new software for H E curriculum areas has been purchased, previewed, or recommended in the past year; and what hardware is used or about to be purchased by H E departments. The survey information as well as the Resource Guide was stored on microcomputer for easy access, cost effectiveness, and reduction of editing time. The annotated bibliography from the initial grant was reviewed by the advisory committee to assess any necessary format changes. The new Resource Guide was then entered into the microcomputer for easy access and integration of additional software programs.

(2) To develop new microcomputer software for the H E curriculum areas by contracting with CCC instructors, consultants and computer programmers based on highest need as established by a survey. The advisory committee reviewed the suggested areas for software development, and decided on three programs to be developed. Contracts made with computer programmers to develop the actual software emphasized user friendliness, that is, easy to use and hard to misuse.

(3) To disseminate the new microcomputer software and Resource Guide to H E departments in CCC, the Chancellor's Office and ERIC.

This grant was implemented through the Northeastern California Higher Education Council (NCHEC), a consortium of Butte, Feather River, Lassen, Shasta, Sierra, Siskiyou, and Yuba Colleges, plus California State University, Chico. All seven community colleges and Chico State offer academic courses in H E. As a group they are committed to improving instruction and keeping abreast of the uses of microcomputers to facilitate such improvement.

RESULTS

The main objectives of this project were trying to close some of the gaps in microcomputer software through further development in H E areas, and identifying and evaluating what already exists and disseminating this information to all California Community colleges.

A. Surveys Taken

The first survey (Appendix A) was sent to all 92 campuses having H E programs, for up-to-date data on microcomputers in their areas. Of the 75 responses, Table 1 (Appendix B) indicates the types of computers available on these campuses. The Apple was by far the most available microcomputer, followed by the TRS80, then Commodore Pet, the IBM PC, and Texas Instruments.

Table 2 (Appendix C) summarizes software needs by rank order based on the curricumum area's first, second, and third choices. Foods and nutrition ranked first, then child development, textiles and clothing, interior design and housing, resource management, and, last family life and parenting. It was interesting to see that each curriculum area tended to suggest its own area as being in most need.

Table 3 (Appendix D) indicates primary reasons given for currently using microcomputers in H E education. Not understanding possible applications, followed by lack of administrative support and lack of an introduction to the microcomputer were common reasons presented. Many still suggested that the microcomputer was not useful or appropriate in the field or home. Lack of time and money, inadequate software and no computer background were other reasons cited.

For suggestions in developing software, it seemed many were unaware of what was already available. For example, diet analysis programs were suggested even though there are probably 25 diet analysis programs readily accessible to them. Each time a potentially good idea was suggested, an already developed program was found or one with very similar content. A logical step was to first survey the broad field for available software and then make a decision for new software development.

B. Resource Guide Development

A second survey was sent to community college and four-year university instructors of H E, commercial vendors,

software and hardware manufacturers, computer users' groups, and other relevant resources for compiling data for the Resource Guide. Whenever preview copies were available, they were reviewed. Otherwise, the annotations were based on information supplied by the author, users, and publishers.

A format was developed using Data Factory as the data base because of its flexibility. Additions, deletions, and corrections can readily be made to the Resource Guide listing new software, books, periodicals, change of source addresses, and update of other relevant information. This format was used in development of a similar resource guide for agriculture programs in California and found to be well received by the users.

The Resource Guide (Appendix F includes sample pages) is meant to be a reference to the H E instructor in order to accomplish the project goals stated earlier. It provides resources for personal computer and peripheral manufacturers, periodicals, and books. A user glossary is included as a handy review of terminology. The bulk of the Resource Guide consists of current available software for the specific and general content areas of H E.

The data base of software has been cross-referenced so that the user may locate appropriate software through a search of subject matter (e.g., foods or clothing), source (e.g., Dynacomp or Krell), necessary hardware and operating system (e.g., Apple IIe or TRS80, DOS 3.3, or Pro DOS), or addresses of vendors.

A deliberate attempt was made not to include evaluations of the software because of high subjectivity of most reviewers. Rather, all particulars of the software itself are included so that users could judge for themselves. For example, under the heading "Learning Time to Become Competent", if .5 hour is required, a beginning computer user might consider this program. If instead, 5-10 hours are required, a beginner may choose to select another less time consuming piece of software. "Intended Users" information indicates the level the programmer focused on, in developing the software--some software has a broader range, others are limited. The user would determine appropriate level based on individual usage.

Interpretation of "Number of Current Users" would suggest that the larger the number, the more the software has been utilized in the field and been proven satisfactory--debugged, that is. A new program without any current users, would then be a chance or choice for the user to take, based on other valid and appropriate criteria.

"Hardware and Operating System Requirements" would assist users in knowing what equipment is essential in running the software. If the user's institution does not currently have the hardware and is unable to purchase it soon, then that software is not a viable choice. On the other hand, if the software is high caliber and exceptionally relevant to the user's needs, then hardware and operating system requirements would be an excellent justification for future purchase.

"Documentation" is a practical benefit to users. Manuals, curriculum materials, teacher guides, on-line interaction with help menus, all determine the user friendliness of software.

The user is not limited to knowing the "Language" used by the software developer, since this is an internal variable. However, if the user is familiar with the language, then adaptations and modifications can be directly made to the software if it is not commercially write-protected.

"Costs" vary drastically and are not necessarily related to quality. If previews are available, that is certainly an asset in determining purchase and ultimate use. The latest revision of field tested software is generally preferable.

C. Software Development

After extensive review of what software really was available, the second phase of the grant--software development--began. The instructional coordinator for computer education was consultant for the grant and directly supervised the programming. Computer science senior and graduate students were hired to actually program. Faculty, advisory members and subject matter experts in the CCC cooperatively worked with the programmers in development of the software as well as related curriculum materials. An early childhood education program for administration of a nursery school resulting in a realistic profit loss statement was first developed. Emphasis was placed on a student being able to implement theoretical statements in running a nursery school while simulating a variety of "what if" case studies.

A second program primarily to be used in clothing and textiles and interior design areas highlights the graphic capabilities of microcomputers. A quilting design can be created on the screen, preferably a color monitor, as one square of the quilt. The program then expands

the design in a variety of angles in order for the user to visualize the entire quilt pattern. Backgrounds and patterning colors can be changed at will. The pattern can also be saved on hard copy--in color if an appropriate printer is available or in shades of black and white on a typical less expensive model.

A third program visually compares nutritional recommendations (RDA) for individuals using "PFS File Report and Graph" as the basis for the software.

SUMMARY AND RECOMMENDATIONS

Generally it was found that much more time was spent on programming than was originally planned. Software is like an onion: it comes in layers of complexity. The more layers there are, the more peeling the onion makes one cry. For every 25 percent added to the complexity of the problem, another 100 percent is added to the complexity of the software which solves the problem. It is very easy to underestimate the impact of adding "just a little something" to already complex software. Faculty, advisory members and field test users who were more familiar with programmming were amenable and reasonable with requests for change.

A third grant funded for the CCC will follow up on the previous grants. More advanced level workshops that include hands-on experiences with data bases, word processors, and spread sheets will be offered throughout the state and include a more extensive field testing of the newly developed software. Updating the Resource Guide will be continued from input of the participants. A more selective screening should take place as a result of usage in the field.

Futurists generally encourage professions to be on the leading edge. And yet, in purchase of equipment and software, it perhaps is better to be somewhat cautious for several reasons: cost comes down drastically; bugs are worked out, mistakes are corrected, complications are simplified; and if something is a fad or is more trouble using than another media, the "time test" will suffice. Augmenting classic software through relevant applications in H E such as templates in using PFS or Visicalc are more efficient ways of implementing computers in the classroom. Developing appropriate and creative curriculum materials to enhance already existing software would also be valuable.

In summary, the home economist has the opportunity to integrate the computer and other forms of technology into the classroom and home in many ways. The future belongs to those who plan for it--improvement of the quality of life depends on those plans. And the home economist generates them. The challenge now is to BE that "Promising Future in Home Economics".

APPENDIX A--Transparency Only

APPENDIX B:

TABLE 1. Type of Microcomputers Available at Community Colleges

Microcomputer	Number Available
Apple	47
Radio Shack	14
Commodore	7
IBM PC	6
Texas Instruments	2
Other	9
Overall	85

APPENDIX C:

Table 2. Software Need by Curriculum Area

Curriculum Area	Choice			
	1st	2nd	3rd	Total
Foods and Nutrition	36	12	6	54
Child Development	12	9	13	34
Family Life and Parenting	2	4	4	10
Textiles and Clothing	7	8	11	26
Resource Management	3	18	2	23
Housing and Interior Design	3	10	13	26

Table 2 Summary:

In rank order:

Foods and Nutrition	54
Child Development	34
Textiles and Clothing	26
Interior Design and Housing	26
Resource Management	23
Family Life and Parenting	10

Table 3. Reasons Microcomputers are not Currently Used for Consumer and Homemaking Education

Reason	Number Respondents
Have a fear of the machine itself	1
Lack an introduction to the microcomputer	9
Do not understand the possible applications	14
Lack administrative support	10
Feel it is not useful in the field or at home	6
Feel it is a fad or trend	1
Other reasons:	32
Inadequate software	
Lack of money	
Lack of time	
Not available in department	
Lack of microcomputer	
No computer background	
Not appropriate for field	

APPENDIX E--Transparency Only

APPENDIX F--Resource Guide.

Software: How Do You Choose?

Carol Byrd-Bredbenner, Montclair State College

Suzanne Pelican, Pennsylvania State University

Selecting software that is compatible with your computer and that will also provide a meaningful learning experience for students and/or expedite work done at home or on the job is a complex and, all too often, frustrating experience. With the increasing amount of software appearing on the market, the need for objective evaluation of software increases. In fact, the complex and time-consuming process of identifying and evaluating software is probably the greatest obstacle nutrition educators face as they incorporate computers into their activities. Some software is effective, but some may hinder more than help. To aid nutrition educators with this task, the entire Reviewed Materials section of this issue is devoted to a review of software that may have applicability to nutrition educators.

Although we hope that the reviews in this issue will provide guidance in software selection, we recommend that educators preview programs before making final selections. However, we recognize that software is generally unavailable for preview; thus, access to reviews by fellow nutrition professionals becomes increasingly important.

To obtain software for review we compiled a list of all the sources of nutrition-related software of which we were aware or that we were alerted to by our colleagues. Over 200 sources were contacted and asked to submit their software for review. Approximately 50 submitted software programs. Each program was sent to reviewers who had volunteered to assist with this issue; their reviews appear in this section.

The reviewers used a standardized instrument that we developed for evaluating these programs. Instrument development involved collecting and examining a number of existing software evaluation instruments. Although none of these fully met our needs, they helped us to identify many points worth considering when selecting software. From these we derived two preliminary instruments, one specifically for nutrition education software, and another for nutrient analysis software. The differing purposes and attributes of educational and nutrient analysis software indicated the need for separate instruments. We also wrote a guide to accompany each instrument. We felt that a guide was essential if evaluator responses were to be standardized.

We mailed these preliminary instruments and guides to the group of approximately 50 nutrition educators who had volunteered to review software for the Journal. They examined each instrument and its guide for clarity and completeness and made recommendations for improvement. We then made revisions based on these recommendations. Reviewers used these revised instruments to evaluate the software. The information contained in these instruments and guides appears below; for a variety of reasons (including readers' convenience, space limitations, and avoidance of

repetition) we modified the format slightly for inclusion here. For example, because information elicited in sections I-III was the same on both forms and guides, we have listed this general information only once. Also, because the documentation review procedure is the same for both nutrition education and nutrient analysis software, we list the documentation criteria and definitions in the nutrition education instrument which appears first, and then refer readers to that instrument when they reach the documentation review section of the nutrient analysis instrument.

The readers should be aware of the basic instructions we gave reviewers regarding these instruments. First, we acknowledged that some of the questions and evaluation criteria were subject to varying personal interpretation, but we encouraged reviewers to focus on the basic sense behind each criterion as understood by most nutritionists and educators. Second, we pointed out that an italicized term in either instrument was explained in the corresponding guide. Third, we requested that reviewers respond to questions/criteria related to target audiences in terms of the audience for which the reviewers (not necessarily the developers) felt the software was most appropriate. Fourth, we alerted the reviewers to the interchangeable use of the terms "software" and "program". Fifth, we asked that, whenever possible, reviewers of nutrition education software observe while someone from the target audience tried the program. Finally, to help standardize the evaluation and comparison of nutrient analysis software, we provided reviewers with a 20-item sample food list.

We present this information so that readers have access to the criteria that reviewers used in evaluating the software. We also hope readers can use these instruments and guides to assist them in selecting software not reviewed here and to aid them in developing useful software for nutrition education and nutrient analysis purposes. The instruments are comprehensive, but the importance of any one item on the instruments will depend on the intended use of the software and the needs of the particular educational setting. We encourage modifications to adapt these instruments to meet specific needs.

Taken from Software:How Do You Choose?, Journal of Nutrition Education, June, 1984, Vol. 16, No. 2. Permission to reprint requested.

State of the Art: Microcomputers In Home Economics Higher Education

Peggy S. Meszaros, Ph.D.-Oklahoma State University

M. Sue Stanley, M.S. - Oklahoma State University

ABSTRACT

The purpose of this national study was to determine (a) how microcomputers are being used in home economics higher education resident instruction, research, and extension/public service; (b) what development and use of software has taken place; and (c) what are the needed services from a proposed national home economics microcomputer clearinghouse. The survey instrument was a 23 item questionnaire mailed to members of the Association of Administrators of Home Economics and the National Council of Administrators of Home Economics with a 53% return. Frequency counts and percentages were used to analyze the data.

Findings indicate individual faculty development and campus-wide efforts dominate unit planning for microcomputer use. A majority of faculty and students have easy access to microcomputer laboratories at no charge. Problem solving and simulations are the most frequently used undergraduate and graduate resident instructional techniques. Data base management and statistical computations predominate for research uses. Diet analyses and consumer economics software programs have been most helpful to extension/public service respondents. The areas of food and nutrition and consumer economics lead in software development and use. The need for a clearinghouse was supported with evaluation of software, annotated bibliography, and software development the most desired services.

There can be no doubt that our world is transitioning from an industrial age to an information age. Evans (1979) has predicted that the world is on the verge of the Computer Revolution - a revolution even more widespread than the Industrial Revolution. The proliferation of inexpensive and increasingly smaller microcomputers will radically alter all aspects of our lives in the next few decades. The effects are already being felt in education.

The microcomputer is revolutionizing higher education. Two languages are now essential to survival: English and Computer (Naisbitt, 1982). The integration of microcomputers into the classroom has the "potential to dramatically improve teaching and learning" (Meroz, 1984). Computers have the capacity for concentrating student attention which allows for improved achievement (Melmed, 1984). The same potential for revolution exists in research and extension efforts of higher education.

What progress has been made in integrating microcomputers into college and university programs? Few studies exist documenting the higher education use of microcomputers in resident instruction, research, or extension/public service. No study could be located addressing the specific use of microcomputers in home economics higher education programs.

The purpose of this national study was to determine (a) how microcomputers are being used in home economics higher education resident instruction, research, and extension/public service; (b) what development and use of software packages has taken place in resident instruction, research, and extension/public service; and (c) what are the needed services from a proposed national home economics micro-

computer clearinghouse.

METHOD

Subjects and Procedure

Two hundred seventy nine surveys were mailed to members of the Association of Administrators of Home Economics and the National Council of Administrators of Home Economics. Follow up included a postcard reminder and phone call. One hundred forty eight usable surveys were returned for a response rate of 53%. The survey respondents represented 57 units accredited by the American Home Economics Association (AHEA) and 113 Agency Member Units of AHEA. A total of 117 public and 31 private institutions in 42 states and Puerto Rico were represented in the survey responses.

The instrument was a 23 item questionnaire designed to be answered in four parts: General, Resident Instruction, Research, and Extension/Public Service. Items included questions on unit planning for the integration of microcomputers, available hardware, software sources, use of the microcomputer in resident instruction, research, and extension/public service programming, and a potential microcomputer clearinghouse needs assessment. In large units the Dean or Director completed the General Section with Assistant/Associate Deans and Directors completing their assigned areas of resident instruction, research, or extension/public service. In small units one administrator completed the entire survey. Frequency counts and percentages were used to analyze the data.

RESULTS

The results of the national survey will be reported by the research questions.

Question One: How are microcomputers being used in home economics higher education resident instruction, research, and extension/public service?

One hundred eight units (73%) are approaching the use of microcomputers through faculty development. Eighty units (54%) are also participating in campus-wide cooperative efforts. One hundred ten units (75%) have a long or short range plan for the use of microcomputers and 83 (56%) have a specific faculty development plan.

One hundred thirty three (90%) of the units responding indicated students and faculty have access to a microcomputer laboratory. Ninety three of the units (63%) have labs located in either the college or university; 14 (10%) are in departments only. Nine units (6%) have laboratories located in the department, college and university. One hundred sixteen units indicated a total of 2332 microcomputers available to faculty and/or students. The three brands most commonly listed were Apple, Radio Shack and IBM.

Students in the 1984-85 school year are not required to own or rent a microcomputer in 142 (96%) of the units surveyed. One hundred thirty one units (89%) do not require students to pay a fee to use the microcomputer laboratory. Ninety nine units (67%) have no programmer available and 100 units (68%) have no system analyst.

Microcomputer publications listed as most helpful to unit administrators included Personal Computing (11), InfoWorld (9), Family Computing (9), Popular Computing (8), Byte (8), and Creative Computing (8). The Home Economists's Computer Newsletter was listed as helpful by 7 administrators. Catalogs such as Strictly Software, books and miscellaneous publications were also listed.

When asked about present uses of microcomputers in resident instruction, problem solving (52%) and simulation (32%) were Checked most often at the undergraduate level. At the graduate level, problem solving (33%) and simulation (21%) were also checked most often.

Seventy eight units (53%) are using microcomputers in research for database management, 66 units (45%) for statistical computations and analyses, 48 units (33%) for word processing, 45 units (31%) for literature searches, 39 units (27%) for instruction or research techniques, 37 units (25%) for graphics and 28 units (19%) for software development. A total of 72 research projects currently in progress or completed within the last three years focusing on the use of microcomputers were listed.

Question Two: What development and use of software packages has taken place in resident instruction, research, and extension/public service?

The major sources of software used in the resident instruction program appear to be commercially developed packages (84%) and packages developed by faculty and students (31%). Respondents indicated the development of software packages in food and nutrition (20), consumer economics (16), housing (7), textiles and merchandising (6), interior design (2), family relations and child development (2), and 6 software packages covering miscellaneous topics such as resume writing and student record keeping.

Research faculty have developed soft-

ware packages primarily in food and nutrition (11), followed by family relations and child development (7), housing (1) and historic costume (1).

Extension/public service respondents reported the production of 5 microcomputer newsletters and development of software packages in consumer economics (33), food and nutrition (18), family relations and child development (3), clothing, textiles, and merchandising (1), and housing (1). Twenty nine extension/public service respondents identified 79 software packages produced commercially and by faculty as being most helpful to them in programming.

Question Three: What are the needed services from a proposed national home economics microcomputer clearinghouse?

When asked if they would find it helpful to receive a national newsletter on the use of microcomputers in home economics 129 respondents (87%) said yes. Topics suggested ranged from administrative questions such as the operation of computer laboratories to software and hardware updates. Specific subject matter software evaluations and research topics were also suggested.

Administrators and resident instruction respondents ranked the top three services most useful from a national clearinghouse for microcomputer use in home economics as the evaluation of software, an annotated bibliography of software, and the production of software.

Research respondents ranked software inventory first, an annotated bibliography next, and literature search third. The development of software followed.

Extension/public service respondents ranked the top three services needed as an annotated bibliography of software, evaluation of software, and software development.

CONCLUSIONS

There is evidence of long and short range planning for the use of microcomputers in a majority of the home economics units responding. Faculty development and campus-wide cooperative efforts are the major techniques used.

A majority of faculty and students have easy access to microcomputer laboratories at no charge. Apple, Radio Shack, and IBM microcomputers predominate. The services of a programmer or system analyst are not generally available.

Problem solving and simulation are the most frequently used resident instructional techniques for integrating microcomputers into undergraduate and graduate programming. The major sources of software in the resident instruction program are commercially developed. Software developed by faculty and/or students has been concentrated in the areas of food and nutrition and consumer economics.

Researchers are using microcomputers primarily for database management, statistical computations and analyses, and word processing. Software developed has been in the areas of foods and nutrition and family relations and child development. There is evidence of research productivity judging from the listing of microcomputer research projects currently in progress or completed within the last three years.

Extension/public service respondents indicate production of software primarily in the areas of consumer economics and food and nutrition. Extensive use of commercially developed software is evident.

Priority services desired from a national clearinghouse were software evaluation, software development, and an annotated bibliography of software. A national newsletter on the use of microcomputers in home economics was strongly endorsed.

REFERENCES

Evans, C. (1979). The Micro millennium. New York: Viking Press.

Melmed, A.S. (1984). Educational productivity, the teacher and technology. T.H.E Journal, 11, (7), 78-82.

Meroz, H. (1984). Realizing the educational potential of microcomputers. T.H.E Journal, 11, (8), 107-110.

Naisbitt, J. (1982). Megatrends: Ten new directions transforming our lives. New York: Warner Bocks, Inc.

MICROCOMPUTERS & RESEARCHERS:

INFORMATION RETRIEVAL

Frederick G. Knirk

University of Southern California

ABSTRACT

The purpose of this paper is to provide information regarding the availability, usefulness, and procedures for using microcomputers in educational settings. This will be done by examining: 1) general education/research oriented information services, 2) specific software programs which can aid a search (e.g., "In-Search", Menlo Corp.), and 3) specific databases (particularly those relevant to computer-oriented educators/trainers). The procedures for using them will also be explored.

The database networks included are those used to rapidly and effectively search appropriate research literature: Bibliographic Research Service (and After Dark), Dialog (and Knowledge Index). These databases comprise many already known to professionals in their print-based equivalents: Psychological Abstracts, Dissertation Abstracts, Foundation Directory, ERIC, Index to Health and Safety Education, Book Review Index, Career Placement Registry, Official Airlines Guide, and so on. These databases are instantly and conveniently available from our office desks providing the availability of a desktop microcomputer or terminal, modem and appropriate software. This process can dramatically reduce the time required to retrieve relevant and timely information needed for an idea or literature search. Equipment selection, database selection, software selection and procedural search issues will each be discussed.

INTRODUCTION

Working professionals constantly require immediate access to relevant information involving their current interests. The computer center manager will often provide these user services directly through the computer center. In the past few years dramatic progress has occurred in providing on-line information storage and retrieval service. Information can now be stored in a more compact and less expensive manner and it is now possible to retrieve more relevant and current abstracts and documents.

The information networks include DIALOG, BIBLIOGRAPHIC RESEARCH SERVICE, THE SOURCE, and COMPUSERVE. The databases they include were often taken from print-based third-party equivalents; Environmental Health and Pollution Control, Environmental Period Bibliography, Environmental Abstracts, Index to Health and Safety Education, Index to Environmental Studies, Chemical Abstracts, Congressional Record, Psychological Abstracts, Academic American Encyclopedia, Book Review Index, Career Placement Registry, Dissertation Abstracts, Educational Resources Information Center, Foundation Directory, Dow-Jones, Standard and Poor's News, Official Airlines Guide and so on. Many of the same data sources educators have been using are now available from the micro-computer or computer terminal on our desks. The information in thsse eletronic "libraries" is current; usually up-dated on a weekly or monthly basis and often is less costly per use than the paper alternative. You usually connect your terminal or computer to these networks by dialing a local telephone number.

DATABASE AVAILABILITY & COSTS

The costs of these networks are generally minimal. DIALOG and BIBLIOGRAPHIC

RETRIEVAL SERVICE (BRS) are professionally oriented databases and provide almost identical services. These include most of the files mentioned above. There is no cost involved in obtaining an account on these systems. In fact, the subscriber will often be provided with free time to learn the system. It will probably cost between $50 and $500 to get enough documentation on the system to enable you, your secretary, your customer service representative, or "librarian" to effectively search the databases. For about $200 one can attend a day-and-a-half seminar which DIALOG offers in many cities throughout the world. The seminars and documentation are well done. There is no monthly minimum charge; it varies according to the amount of time the system is used. The costs of the databases range from about $25 per hour plus printout charges (if you request DIALOG to print the files out for you and mail them to you) of .10 per file for some of the government supported files like ERIC to over $100 per hour plus printout costs. In addition, you may expect data communications (telephone) charges of $6 (Dialnet) or $8 (Telenet, Tymnet, Uninet) per hour.

For the personal computer user who would like to use these databases, Dialog has a "Knowledge Index" service. This offers a selected group of files (ERIC, MEDLINE, GPO PUBLICATIONS, STANDARD & POOR'S NEWS, PSYCHINFO, MAGAZINE INDEX, and others) at a $24/hour rate (which includes communications charges) with a $35 initial connect charge (two free hours are credited). Search is done with English words as "begin", "find", "display", "order", and "logoff". You search for keywords which may be linked together with "and", "or", or "not"; example--FIND LEARNING AND CLASSROOM AND (COLOR OR LIGHT) NOT CONTRAST. This service is available after 6 in the evenings, all day Saturday and from 3 PM on Sunday. Billing is done through major credit cards. BRS also makes 20 of its 50 databases available to the personal computer user through "BRS After Dark". For this service there is a $50 initiation fee and a $12 minimum monthly charge plus any connect-time charges billed to your credit card. The per hour costs for the databases range from 6 to 15 dollars per hour.

COMPUSERVE and the DOW JONES NEWS/RETRIEVAL are more personally than professionally oriented. The passwords, telephone numbers, and necessary documentation for both databases are all available for a one-time charge of under $30. COMPUSERVE costs about $22 per hour for 300 Baud service (between 5 AM and 6 PM, Monday through Friday), and $5 per hour (for non-prime time use) plus, in some cases, about $2 per hour for telephone costs (Telenet or Tymnet). CompuServe has over 140,000 subscribers. This database will allow you to access the Official Airlines Guide, stock market data (Standard and Poor's, Value Line Data Base II), Federal Reports, Federal regulations, Federal publications, an electronic mail service (more on this later), games and other consumer databases.

The DOW JONES NEWS/RETRIEVAL system costs about .15 per minute for historical and current (delayed 15 minutes) quotations, about .20 per minute for an electronic edition of the Wall Street Journal or the Dow Jones News., about .30 per minute for the Academic American Encyclopedia (about 9 million words, updated every 6 months) to about .90 per minute plus additional charges for its financial and investment services databases.

The SOURCE is a subsidiary of The Reader's Digest and offers services similar to CompuServe. The initial cost for the password and documentation is $49.95. During the day the cost for 300 Baud service is about $20.75 per hour (7 AM to 6 PM) but is reduced to $7.75 in non-prime time. The SOURCE has a monthly minimum of $10. The 1200 Baud service is $5 more during prime time and $3 more in non-prime time. By mid-1982 the Source had about 17,000 subscribers, mostly non-business users. Included in this network is a database listing airline schedules (lists entire domestic and foreign airline schedules for flights originating in North America) which is a value-added service costing $39.75/hour or $34.75 in non-prime time, the Mobil Restaurant Guide, stock market information, an electronic mail service, a word processing service, foreign language lessons (in Spanish, French, Latin, Greek, Italian, German, and Esperanto), and more.

Both COMPUSERVE and SOURCE offer weather and news information. CompuServe supplies up-to-the-hour weather data, read directly from the National Oceanic and Atmospheric Administration computers. CompuServe receives all its news from the Associated Press. Reports are updated whenever possible after the information is received. CompuServe primarily provides menus with story titles to aid in the search for relevant information. The Source offers weather and news from the United Press International services. Users enter a keyword to search all stories in a category.

In general, access to information on

these information systems is simple and relatively inexpensive. After obtaining a password, a telephone number, and basic documentation on how the system works, you are ready to access the data. The costs range from about $10 per hour to over $100 per hour depending on the information system you are using, the time of day you intend to use the system and the database you select to search.

A summary of database costs together with database addresses and telephone numbers are included at the end of this paper.

SEARCH STRATEGIES

Professionally oriented databases like DIALOG use a keyword oriented data structure involving Boolian arguments (and, or, not) to relate the keywords/descriptors to each other in a search. Let us assume, for example, we have three related concepts to explore: 1) effects of light intensity, 2) reaction times to panel displays, 3) on the elderly. You could combine these three concepts in a single search to obtain only references including all three of these concepts/descriptors. Use a thesaurus of descriptors, if one is available for the database to identify the terms used by the database abstractors to classify documents. Then enter the appropriate file and conduct the search. The statement to the system might appear as: "ss light and reaction-(w)times and elderly". This search would yield only reports, publications or articles with these specific words. You might wish to search synonyms as well. For example, with regard to light intensity, these keywords could also be entered: light radiation, glare, lighting, visual environment, human factors engineering, light design, contrast or light(w)intensity. Selecting some of the more relevant descriptors, one might use this search strategy: "ss (light or light(w)intensity or visual environment) and (reaction(w)time? or response(w)latency or perceptual(w)motor (w)coordination) and (elderly or older(w) adults or senior (w)citizens)". A listing of the number of publications available for each heading would be provided and the user could then decide if the search should still be narrowed or expanded. Note that the "ss" simply tells the computer to select the information published between specific time periods, search by author, or search by titles only. Finally, a database user can type the individual abstracts out locally or have DIALOG print them out and mail them to you.

Many files can be searched by limiting the dates or language of the publications. To limit the search in Dialog one would enter PY=1980:PY=1985 to search for documents from 1980 through 1985. To limit the search to English language materials in PsychInfo (Psychological Abstracts), enter LIMITALL/ENG when you begin the search.

Searching COMPUSERVE or the SOURCE involves either using a menu to transfer from one database to another or a page numbering scheme. Once in a database, one can choose from among the title of, say, the AP wireservice stories to select a story to read. This is especially rewarding using the DOW JONES network to access the Wall Street Journal. The Dow Jones News/Retrieval Service has about 30,000 users. The cost for this 300 Baud service is about $60 per hour during prime-time and $12 an hour otherwise. 1200 Baud service is about 50% higher than 300 Baud service.

The DOW JONES information network has an encyclopedia online. This network has the entire 20 volume Academic American Encyclopedia available to its users. It has over 28,000 articles and is updated twice a year by Grolier Inc. The encyclopedia costs 60 cents per minute during prime time and 30 cents during non-prime time.

MICROCOMPUTER SOFTWARE SEARCH ASSISTANCE

The personal comptuer user can be greatly aided by software programs designed to assist in a Dialog search. In-Search (Menlo Corporation, 408/986-1200, $399) is the best. The purpose of In-Search is to make searches simpler and thus less costly for Dialog users. It does this by helping a user to decide on a search strategy before signing onto Dialog. This "access software" helps to dial the access number, log onto Dialog, access the desired database, order the predetermined search, allow the user to modify the procedure and order a "type" or "printout". It is difficult to bomb a session; the program is smart enough to ask you if you really want to do it. The producers suggest a 15 to 20 percent reduction in search time, thus costs, when this program is used.

DATABASE AND INFORMATION SERVICES

The following is a list of databases useful to educators and trainers:

BRS, Bibliographic Retreival Services

Inc., 1200 Route 7, Latham, NY 12110, 518/783-1161. Access to major databases, primarily for libraries, MEDLARS, Psych Abstracts, Books in Print. About $35 an hour depending on files used plus royalty and telecommunication fees; annual subscription payment of $50.

BRS After Dark (Bibliographic Retrieval Service), 1200 Route 7, Latham, NY 12110, 800/833-4707. Educational databases as: ERIC, School Practices Information File, Exceptional Child Education Resources, Bilingual Education Bibliographic Abstracts, Mental Measurements Yearbook, Academic American Encyclopedia. $150 subscription fee $18/hour charge plus royalties and print charges. $18 mimimum monthly charge.

CompuServe Information Service, 5000 Arlington Center Blvd., Columbus, OH 43220; 614/457-8600. Costs range from $12.50 daytime or $6 per hour for off-hour use (plus communication charges if any) for Associated Press, Official Airlines Guide, World Book Encyclopedia and other databases of interest to home users. Initial fee is $39.95 with no monthly minimum.

Delphi, General Videotex Corporation, 3 Blackstone St., Cambridge, MA 02139, 800/544-4005, includes the Official Airline Guide, computer shopping, games, sends letters via ECOM, etc. Access to Dialog, Source and CompuServe mail services. You can get a starter kit for $30 which includes an hour of free time. It costs $6 an hour in the evening and $16 during the day with no Tymnet charge, no monthly minimum and no premium for 1200 Baud operation.

Dialog Information Retrieval Service, 3460 Hillview Ave., Palo Alto, CA 94304, 415/858-2700. Largest system with over 40 million records. Includes ERIC, Psych Abstracts, NASA, etc. Costs from $24 an hour for ERIC to over $100 plus $8 per hour for Tymnet charges. A ten minute search could range from $5 to $15 per ten minutes, depending on database. No minimum monthly fee.

Dow Jones News/Retrieval, Dow Jones & Co., Inc., P.O. Box 300, Princeton, NJ 08540, 800/257-5114, 609/452-1511 is a financially oriented database.

GTE Telenet Medical Information Network, 8229 Boone Blvd., Vienna, VA 22180, 703/442-1000 is designed to provide clinical information to physicians, nurses, etc. $100 initiation fee plus $27 per hour prime time and $7 non-prime time.

Knowlege Index (Dialog's after 6 PM service), Lockheed Corporation, 3460 Hillview Avenue, Palo Alto, CA 94304, 800/982-5838 (CA) or 800/833-4707. Includes many of Dialog's most popular databases. Initial cost of $35 includes two hours free time and necessary documentation. $24 per hour (either 300 or 1200 Baud) with no monthly minimum. Access via Tymnet, Telenet, Uninet, or Dialnet. Billed to credit card.

Lexis/Nexis, Mead Data Central, Dayton, OH 800/227-0597. Lexis is an online legal database. Nexis offers periodical information. Inlcudes the Encyclopedia Britannica (43 million words), online and abstracts of the New York Times, etc. A 1½ minute search may produce over 150 "hits" and cost about $10.

OAG, Official Airline Guides, Inc., Oak Brook, IL. $50 initial fee with an average charge of $2.25 per session.

SDC Search Service, 2500 Colorado Ave., Santa Monica, CA 90406. Databases in business, economics, chemistry, engineering, electronics, energy, government, industry, science and technology and social sciences. Rates, not available.

The Source, 1616 Anderson Road, McLean, VA 22102, 800/821-6666, 703/821-6660. Geared to home and business use. Includes UPI news, Unistox market information. $49.95 registration fee; $20.75 per hour prime time, and $7.75 non-prime time with a $10/mo. minimum charge. Fee for 1200 Baud service of $3 or $5.

WESTLAW. Electronic legal research library. Available through Radio Shack. 1200 Baud Modem Required. $249 plus user charges.

EQUIPMENT AND TELEPHONE NETWORK REQUIREMENTS

The equipment needed for accessing these databases include 1) either a computer terminal or a microcomputer with a software modem program, 2) either an acoustic coupler or direct-connect modem operating in (at least) originate, full duplex mode at either 300 (Bell 103 compatible) or 1200 Baud (Bell 212A or Racal-Vadic compatible) and be FCC approved. In addition you must contact the database personnel to obtain an account number, password, and a list of local telephone numbers to access a packet-swithcing network such as Telenet, Tymnet or Uninet.

To locate the local number for TELENET call 800/336-0437 (in Virginia 800/572-0408).

To locate the local number for TYMNET call 800/336-0149 in the East and 800/323-7389 in the West. Ask for a free Tymnet Online Newsletter. For information log on, type a, at "please log in" type INFORMATION;;. If a prompt "@" calls for Project Code: hit (Return), to a prompt asking for an item number type ? for the

14-item menu. To get a demonstration of the SOURCE using TYMNET, dial the local Tymnet number, type "A" for type of terminal if 300 Baud, type "SOURCE12;PRIM" at "Please Log In:". When the "+" sign appears hit (enter)-/(return). Type "ID STCOMO DEALER" for your ID number and password. This will provide a 15 minute tour of the Source. If you use TELENET, at the @ sign type C 30137, then continue as above.

REFERENCES

Computer-Readable Databases: A Directory and Data Sourcebook. White Plains, NY: Knowledge Industry Publications, Inc.

Direcotry of Online Databases. Santa Monica, CA: Cuadra Associates, Inc. (Quarterly).

Ferrarini, E. M. (1982, September). A wise world awaits your computer. Business Computer Systems, 31-32.

Frenzel, L. E. (1982, June). Learning with micros. Interface Age, 36-37.

Glossbrenner, A. (1983). The Complete Handbook of Personal Computer Communications: Everything You Need to Go Online with the World. NY: St. Martin's Press.

Glossbrenner, A. (1984, October 16). Tricks of the on-line trade. PC 3 (20), 176-192.

Howitt, D., & Weinberger, M. (1984). Databasics: Your Guide to Online Business Information. NY: Garland Publishing.

Information Industry Market Place. (Annual). NY: R. R. Bowker.

Owen, R., & Doroniuk, E. (1982, July). Computer communications using the telephone network. Interface Age, 71-76.

Taylor, J. (1984, October 16) On-line house calls. PC 3(20), 134-141.

Westin, A.F. (1982, July). Home information systems: the privacy debate. Datamation, 100-114.

Effective Manuals Are More Than Content

Patricia Anne Lawson

Microtel Learning Services

ABSTRACT:
The effect of manuals on long-term learning as post-requisite to training cannot be over estimated. As well as readability, effective manual preparation requires careful planning to ensure the appropriateness of the material, legibility, accessiblity and attractive, utilitarian manual design and packaging. This paper provides a comprehensive, annotated list of criteria by which to address these considerations.

Manuals are essential to training, during and after sessions. They give and maintain learner confidence and independence. Their effect on long-term learning cannot be over-estimated. While much has been written about composition and readability, effective manuals are more than content. This is because manual users today are no longer a captive audience. They assume content validity. More than that, they demand appropriateness of material, legibility, accessibility of information, attractive manual design and attractive, utilitarian packaging.

1. Appropriateness of Material

Not all information lends itself to incorporation in a manual. For example, affective learning is better addressed through multi-media sessions. The types of information that are appropriate include explanations, concrete examples, tutorial information, quick path information, job aids, trouble-shooting helps and library information, in other words, the hard core material.

- Explanation - A brief understanding of 'why' something is important or 'how' it relates to the whole promotes retention. Also, brief explanatory paragraphs tie a manual together and maintain flow.
- Concrete Examples demonstrate where readers can apply information and provide a basis for generalization to situations they might encounter.
- Tutorial Information is useful for reinforcement and as an instructional tool. Often the individual a company sends to a course must instruct others. Without good tutorial information, the success of the course can depend solely on that individual's ability to teach. Tutorials take learners step-by-step through a series of written lessons designed to establish a reasonable skill level. The lessons are arranged in a need-to-know order and assume no reader prerequisites.
- Quick-Path Information is the inverse of tutorial information. It assumes that readers have a certain background and are looking for concise information. Whereas tutorial information might take the reader four hours, quick-path information might take half an hour.

- Job-Aids are similar to quick-path information in that they incapsulate important information. Whereas quick-path information is usually explanatory, a job-aid is usually procedural. Job-aids take the form of checklists, diagrams and/or algorithms. They are intended to be removed from the manual and used wherever the learner applies the information learned.
- Trouble-Shooting Helps are similar to job aids except that they deal with exceptions rather than rules. They are best presented as flow charts or algorithms, followed up with checklists and/or explanation.
- Library Information - Often instructors must use less-than-satisfactory existing manuals. Information on where to find information in these manuals and/or an explanation of how to use them together with updated additional information, can often best be accommodated in a separate library document.

2. Legibility

Readability relates to the number of syllables, words and sentences in a section of text; legibility refers to the conventions used in committing the words to paper. Readability ties the organization of the written word to comprehension; legibility ties it to speed and ease of intake. Miles Tinker and followers provide empirically proven rules like:

- Use blocks of capitals as little as possible and never in the text.
- Indent paragraphs; they are a visual cue to a new idea.
- Keep paragraphs short, a maximum of 75 words.
- Use ragged right margins rather than right justification. This avoids awkward word spacing and hyphenations.
- Single space text and lists.
- Group lists into clusters with sub-headings.
- Use black lettering on off-white paper. Reserve one color (only) for specific messages and/or identifiers.
- Only use numbers for a sequence of steps.
- Use arabic rather than roman numerals.
- Use headings and sub-headings rather then the "engineering" convention of number decimal number (4.1.1).

3. Accessibility of Information

Accessibility refers to how quickly readers can find the information they want in a manual. A detailed table of contents only partially meets this need. Good manuals also include a table of contents, preview and summary with every section. A cross-referenced index is also invaluable. If the table of contents and index are roadmaps as to where readers may locate information, tabs, color-coding, headers and footers, and page-numbering are signposts to tell readers where they are and help them access the next signpost.

- Tabs provide physical assistance in getting from one section to another. Further, their labels allow users to access a section without referring to the table of contents.
- Color-Coding - Colored tabs associate a color with certain blocks of information. This color-coding may be echoed in headers or footers, or color bars on the top, side or bottom of individual pages to reinforce location.
- Headers and Footers maintain consistency within sections and also verbally reinforce location.
- Page-Numbering - Pages numbered with the section number followed by the page number numerically tell readers where they are, e.g. 2-43 is page 43 in section 2.

4. Attractive Manual Design

The various types of material appropriate to manuals lend themselves to various modes of presentation. While explanations are best suited to paragraphs of text, job aids may be checklists, diagrams or algorithms. A manual can be or incorporate flip-charts, posters, foldouts, booklets, a series of laminated job cards and so on.

The form of the manual is a design concern. It depends on the kind and amount of information to be included in the manual, as well as what it is to be used for.

The design of the information on the page is as important as the form and must be as carefully designed.

These format considerations include page size; layout; typeface; typesize and spacing; illustrations and color; tables, graphs, diagrams and symbols; and the type of paper.

- Page size depends on the form of the manual and its use. For example, a give-away, for eventual use as a job aid, might best be a pocket-size page with plastic ring binding. If cost is the primary consideration, the standard 8 1/2 x 11, three-hole punched format might be better. Different manuals for the same course need not be the same size although a standardized page size promotes interchangeability of information.
- Layout relates to the position of the written text and artwork on the page, and the amount of white space. Studies indicate that optimum learning results by limiting the width of printed material to 4 to 6 inches and leaving a good margin at the top and bottom. Blocks of text need not be centered but can be shifted left or right. To increase accessibility, text could be moved toward the center of the page to leave the outside margin free for headings or special messages.
- Typeface refers to the style of the lettering. The important factor is the seraph or tail on the letters. Studies link readability to the seraph. The seraph helps distinguish one letter from another. Many learned to read with this typeface.
- Typesize and spacing contribute to the amount of whitespace on the page. Typesize refers to the point value given with typeface, e.g. Helvetica 16 pt. Typesize determines the number of words per line. Spacing determines the number of lines per page. Spacing refers to the amount of white space between lines of type; it can also refer to the amount of space between words. While this is standard for most typeface styles, special requirements may be specified when offset or typeset printing.
- Illustrations and color attract or distract. Illustrations can support text by presenting concrete images to convey abstract ideas and concepts. Illustrations can also communicate visual and spatial concepts as well as several ideas that must be considered at the same time. With text, they need to be as close to it as possible. They can also be used purely as entertainment.
- Color is used to aid instruction or accessibility, or for aesthetic or motivational reasons. Color choice is important; 9% of the population is color-blind. Too much or too many colors set up a dazzling effect and distract.
- The type of paper contributes to the durability of the manual. Thin paper may tear easily and may be too transparent to take printing on both sides of the page. Thick paper adds bulk to the binding process and may make pages difficult to turn or lie flat. Glossy paper looks nice but may be difficult to read because of glare.

5. Attractive, Utilitarian Packaging

While we are admonished never to judge a book by its cover, many judge it by its package. Attractive packaging contributes to ergonometrics and utilitarian packaging to durability. Researchers tell us that packaging increases use and acceptability by readers. Packaging considerations include book size, cover and binding. These are dependent on intended use, user and budget.

- Book Size - Today the convention is that smaller is better.
- The cover attracts the reader and protects the contents. A good design is one which perfects one visual image. The cover identifies the manual; some suggest the title should be readable from across the room. The cover can include pockets to increase its functions.
- The binding holds everything together. It is visual, tactile and functional. There are several options to choose from:
  - Folders and portfolios have no binding as such. They are folded so as to create the illusion of pages.
  - Looseleaf binding is easy to update and stands well on a shelf. It is, however, awkward for frequent use.
  - The standup looseleaf binder can be used like a miniature flip chart for continuous use beside equipment.

- Glued or case binding is difficult to update but looks professional. It is also the most expensive.
- The perfect binding is easier to handle than the looseleaf binding but falls over on a shelf unless packaged in a sleeve.
- Spiral binding allows pages to lie flat which makes it good for continuous use near equipment. Pages can be added or subtracted with some difficulty.
- Saddle-stitched binding is good for small multi-page forms and brochures, and it is inexpensive.
- Stapled packages are appropriate for small updates which may or may not be added to a looseleaf binder.

SUMMARY:

Think about the manuals you have seen and used, especially computer manuals. I suggest their appeal results from the appropriateness of the material to the manual form, its legibility and accessiblity, and its overall design. When they are done well, these considerations are almost transparent. They are, however, the result of careful planning. While they are expensive to implement and their benefits difficult to measure, they are requisite to competitive, learning-effective manual production. Manual "writers" need, at least, to be aware of these factors in order to address them as best they can.

Bibliography:

James Hartley, Designing Instructional Text, Nichols Publishing Co., 1978.

Allan Henderson, "The Care and Feeding of the Non-Captive Reader", Technical Communication, Third Quarter, 1984.

Lowel Thing, "What The Well Dressed Manual Is Wearing", Technical Communication, Third Quarter, 1984.

Miles Tinker, Legibility of Print, University of Iowa Press, 1963.

Marguerite Zientara, "Documentation's Sorry State", Infoworld, Volume 6, Issue 29, July 16, 1984.

CBT Development -- Micro to Mainframe

Travis L. Piper

Creative Approaches, Inc.

ABSTRACT

As corporate Data Processing departments begin to charge users for their services, the cost of using mainframe editors (TSO, CMS, EDGAR, etc.) for CBT development becomes prohibitive. Micro-based word processors, by comparison are fast, powerful, reliable and certainly less expensive.

Creative Course Writer is a series of micro-based programs that help the CBT author to easily develop course content with a word processor and then to edit, list, and prepare it for insertion into the mainframe presentation system (IIPS or Phoenix) while bypassing costly generation by the mainframe authoring system.

Creative Course Writer (or CCW) was developed by Creative Approaches, Inc. of Rochester, New York to meet a client's need for a sophisticated 2-3 hour application simulation course in six weeks. With the capabilities which CCW provides, the simulation course was completed on time and has been described by many as "one of the best CBT courses I've ever seen."

Travis Piper, President

Creative Approaches, Inc.

8 Prince Street

Rochester, New York 14607

(716) 657-7578

The purpose of this paper will be to explain the benefits inherent in the development of computer-based training courseware on a microcomputer using a word processor and then running the resulting text file through a validation program to verify correctness of inputted material and to prepare the resulting file for upload to the mainframe for ultimate execution on the mainframe.

As more and more corporate data centers are beginning to charge user groups for DP services, it is becoming more apparent that the cost of computer-based training authoring by using such editors as TSO, CMS, EDGAR, etc., is approaching the prohibitive mark, not to mention the author's dependence on the mainframe for the system being "up" and the need for good response time.

The subject of this paper is a process developed by Creative Approaches, Inc. of Rochester, NY which makes use of almost any PC-based word processing package for the development of the desired course content. That content is then prepared for viewing on the micro prior to mainframe upload. Upon preparation for upload, the necessary code is automatically added to and removed from the text word processing file to insure proper performance in the mainframe environment. Currently, this environment includes both IBM's Instructional Systems and Goal Systems' PHOENIX.

In spite of the mounting interest in graphics, animation, sound and color for computer-based training delivery, it is apparent that the mainframe environment in the near future, for certain, is going to continue to be a text-based system. This will be true until companies are willing to spend many more dollars on host-accessible terminals or to spend considerable sums on stand-alone training devices, which

The problem of delivering computer-based training through microcomputers is three-fold. First, the logistics problem of physically distributing the course material on floppy disk to remote sites; second, the problem of updatability which causes a redistribution problem of getting the current copies of the course out to the training sites; and third, the physical handling of the disks by students.

It is practically impossible to insure that students at remote sites always have the most current version of a training program since there is no easily manageable way to insure receipt of the training program or incorporation of it into an existing training library and, more importantly, the removal of the old version of the program from the remote site training library.

Also, students are being asked to perform many "computer operator" tasks when taking training at a micro. Enough mainframe terminal CBT students have difficulty simply adjusting their terminal's brightness. This makes a convincing case for avoiding inserting disks, changing disks (necessary with multiple floppy disk courses), not touching certain areas of the disks, and the list goes on. Any act which results in a damaged disk, of course, causes the redistribution problem to arise again.

An option would be to make multiple backup copies of the disks at each remote training site. This approach addresses the backup problem but compounds the updatability issue as now there are even more "old" versions to be called in and more new backup copies to be made by a remote site training coordinator.

To summarize the reasons why microcomputers cannot effectively be used for delivery of computer-based training on a large scale to multiple remote sites, they are: First: logistics and distribution, second: upgradability which is further compounded by the distribution problem again, and third: the physical process of the student operating the microcomputer and handling the disks.

I continue to ask clients if they plan to decentralize their training and run the training on the micro or if they plan to continue to run the training on the mainframe and do the development of the materials on the micro. All of them prefer the latter option but are currently complaining about a lack of products on the market that have all of the capabilities and flexibility of the mainframe system built into them.

Specifically, these include being able to do application simulation and being able to imbed Coursewriter code into the course during development on the micro and then to be able to upload that course to the mainframe.

The product from Creative Approaches which provides all of these capabilities, and more, is called Creative Course Writer or CCW. The product's success is due to its

incorporating all of the mainframe features plus taking full advantage of the power of the microcomputer for such tasks as text-editing, running standalone programs, and, further providing the upload capability to the mainframe. With CCW, a 2-3 hour application simulation course was recently developed for a client in less that 80 hours -- a record-breaking accomplishment.

To use CCW, the CBT author first uses a word processor and simply begins typing screen text just as he or she would like it to appear on the screen. The entire course is developed as a word processing file and contains codes that indicate the beginning of screens of text, the beginning of question text, and the appropriate answer analysis that goes along with a given question. The process further enables the author to enter application simulation code in the form of calling up desired formats, overlaying the format with desired text for predisplay to the student at the time of the screen's first appearing, and then to proceed and do the answer analysis, allowing the author full flexibility of placement of the message text to the student in author-defined windows on the screen. This same process is also possible with regular question screens.

Special characters can be used within the text to indicate that verbiage is supposed to begin on a certain line of the screen rather than having multiple blanks preceding a line. Certain codes can be used to indicate that a given line should be centered when it is output in the final format for mainframe upload. Other commands can be used to indicate that a line should be highlighted, and indeed that any desired combination of attributes be assigned to a line as part of the screen text. The placement of these special codes within a line of text does not in any way impact the alignment of the left margin of the codes as has been a problem with certain mainframe computer-based training systems in the past.

The most desirable feature of CCW is the ability to imbed comments to potential reviewers of the course for explanation as they are reading a printout of the file's content.

Let us assume now that the author has the course, or a portion of it, finished. At this point it is easy for the author to either print the file and make multiple copies of it to be reviewed by subject matter experts or instructional designers; or, the author may let the reviewers review the course file on disk and, instead of making notations on paper, make the changes in the file itself by inserting comments that can be reviewed by the original author and acted upon as appropriate.

The advantage of this process is that the course, at this point, still contains no prefix codes at the beginning of lines that will affect the editing process. The author then, upon receiving the comments back from the reviewers, simply makes use of the word processor to add a line here and there, replace all occurrences of a word or phrase, insert a phrase which will automatically rewrap the remaining lines on the screen, which is very easily done with a word processor.

When the file is in its final form, it can be run through a separate micro-based spelling checker program. This is an excellent way to catch typographical errors, although "The Hair and the Tortoise" would go undetected. The author now executes the CCW editor which performs several tasks.

Edits that are done include the validation and the sequence of the imbedded codes to insure they appear in the course in the proper sequence. As an example, answer analysis cannot be done during a text screen nor can text be overlaid in a screen window when a screen is not currently being displayed on the terminal. Certain edits are performed against the contents of line items on the file such as the line text cannot exceed so many characters in length based upon host terminal model number (mod 5 terminals can be 131 characters where mods 2, 3, or 4 must not be longer than 79), the number of output lines cannot be greater that the screen size. If all of the text passes all edits and validations, the resulting output file is now ready for viewing, listing and upload to the mainframe. If all edits and validations were not passed, a detailed error register of the individual line numbers with problems in them is printed out for the author to make use of in making corrections to the file.

The author can now select the VIEW option of CCW and can see the screen layouts on the micro just as they will appear on the mainframe. All answer analysis codes are displayed and answer replies are positioned on the screen just as they would be on the mainframe. After VIEW, the course can be LISTED.

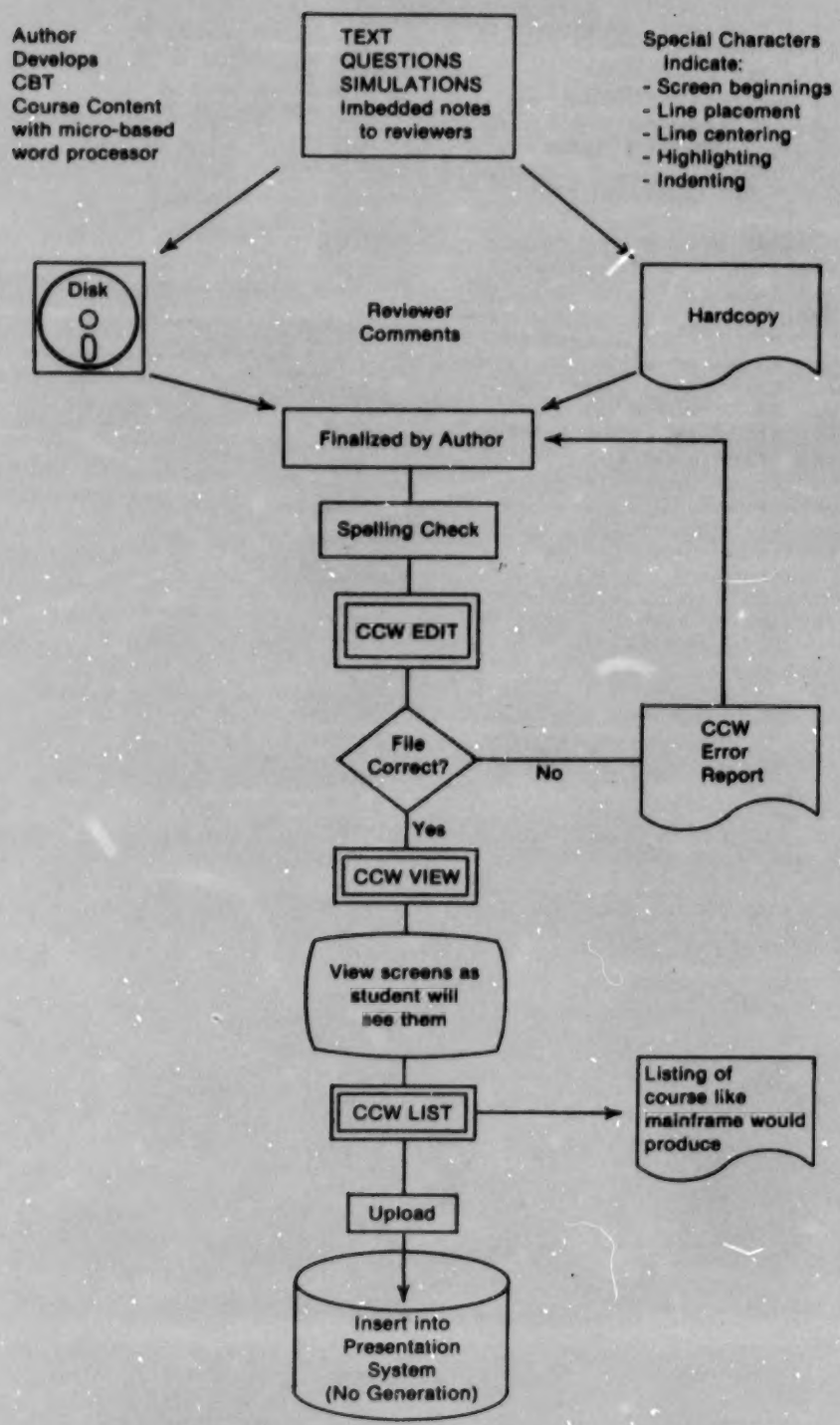
The micro-produced course listing

that is produced from the output file is in the same format as a listing of the course would be if it were to be printed from the mainframe system. This enables the author to see a listing of the course without the need to have the course uploaded and generated before being able to view it.

The final step is to UPLOAD the result of the EDIT program to the mainframe. The code, at this point, is ready for direct insertion into the IS or Phoenix system. No expensive mainframe generation is necessary.

If any subsequent revisions to the course are needed, the affected portion of the word processing file can be isolated, revised, re-edited and uploaded into the presentation system to overlay the outdated code.

CCW provides a revolutionary way to easily develop CBT content on the micro and to ultimately execute the course on the mainframe. As training costs and DP costs continue to rise, CCW offers a cost-effective opportunity to CBT developers to take full advantage of mainframe capabilities using micro tools.


CCW — Micro to Mainframe
Author
Develops
CBT
Course Content
with micro-based
word processor
TEXT
QUESTIONS
SIMULATIONS
Imbedded notes
to reviewers
Special Characters
Indicate:
- Screen beginnings
- Line placement
- Line centering
- Highlighting
- Indenting
Disk
Reviewer
Comments
Hardcopy
Finalized by Author
Spelling Check
CCW EDIT
File
Correct?
No
CCW
Error
Report
Yes
CCW VIEW
View screens as
student will
see them
CCW LIST
Listing of
course like
mainframe would
produce
Upload
Insert into
Presentation
System
(No Generation)

Designing Coursewriter Macros

Timothy D. Taylor

Advanced Systems, Inc.

ABSTRACT

A Coursewriter macro is a tool for authors using IBM's authoring language, the Interactive Instructional Authoring System (IIAS). The macro is a subroutine used by the author during the authoring process for the purpose of generating repeated sequences of similar code. This paper offers some design and development suggestions for authors who are beginning to build and use macros.

What is a macro?

A Coursewriter macro is a subroutine used by an author during the authoring process (not used by a student during course execution) for the purpose of generating repeated sequences of similar Coursewriter code. The Course Structuring Facility (CSF) and Simulation Exercise Facility (SEF) are two sets of Coursewriter macros. You can build your own set if you find that CSF and SEF don't quite accomodate your needs. This paper offers some design suggestions to authors who are involved in the development of Coursewriter macros.

Types of macros based on function

In planning a set of macros, it helps to categorize them into functional groups, such as the following:

1. Framework macros - These set up the structure of the entire course or part of it. A framework macro typically includes code that initializes counters & switches, sets course parameters, and/or creates labels.
2. Text macros - These create screen displays for students to read, or change a display that already appears on the screen.
3. Question macros - Like text macros, these create screen displays, but they also include code for judging answers supplied by students.
4. Branching macros - These use students' performance data as a basis for branching to one location or another within the course.

These categories are not the only conceivable ones, and they might not even be the best ones for you. The categories you use should be determined by the design of the courses you intend to create.

How many macros per screen?

If you use CSF to generate a question screen, you'll use a single macro. But if you use SEF, you'll use several macros to create that single question. So which system is better? It depends.

If you're building a set of macros primarily to make life easier for some authors other than yourself -- authors who don't have your degree of sophistication with Coursewriter -- then one macro per screen is the way to go. You can build "friendlier" macros this way. Also, regardless who the users of the macros are, if you're going to build courses that use only one specific set of question types, rather than many individually architected screens, then one macro per screen makes sense.

On the other hand, if the macros are for your own use or for use by others who speak Hexadecimal and like to play with bits and bytes, then you might consider using smaller macros --ones that serve

only a single function like putting text on the screen or judging a single answer. Using small single-function macros is less "friendly" but gives you considerably more flexibility in designing complicated screen displays and performing complex answer judging. Also, it takes considerably less time to develop this type of macro.

One macro per screen

As an example of the one-macro-per-screen system, consider this hypothetical macro used to generate multiple-choice questions:

```
Macro Name:  mult
Keywords:
  ,title - Text to be centered on line 1
           of the screen.
  ,t03 thru ,t18 - Text to be displayed on
           lines 3 through 18 of the
           screen.
  ,ch    - The number of choices in the
           question.
  ,ans   - The correct answer.  (Only one
           letter is permitted.)
  ,ca    - Words that are equally correct
           and may be typed by the student
           instead of the single correct
           letter.
  ,wa    - Words that are equally wrong
           and may be typed by the student
           instead of single incorrect
           letters.
  ,cf1 thru ,cf3 - Up to 3 lines of
           correct-answer feedback.
  ,wf1 thru ,wf3 - Up to 3 lines of wrong-
           answer feedback.
```

Following is an example of macro call statements used to create a particular multiple-choice question:

```
  mc mult
  ,title=Spanish Vocabulary
  ,t06=The Spanish word for "face" is:
  ,t08=    a) la cama
  ,t09=    b) la cara
  ,t10=    c) la casa
  ,ch=3,ans=b,ca=&cara&,wa=&cama& &casa&
  ,cf1=Right!
  ,wf1=Sorry, the answer is LA CARA.
  gen
```

Multiple macros per screen

As an example of a system using several macros per screen, consider this particular set of macros:

1. frame - To establish the course framework
2. tx - To generate text displays - full screens, text portions of question screens, or feedback messages.
3. aj - To generate answer-judging logic.
4. br - To generate branching logic.

Let's look at just two of the above macros in detail:

```
Macro name:  tx
Keywords:
  ,title - Text to be centered on line 1
           of the screen.  Used only if a
           new text/question screen is
           being created.  (Not used if
           just a feedback message is
           being created.)
  ,t03 thru ,t24 - Text to be displayed on
           lines 3 thru 24 of the screen.
  ,row   - The number of the row reserved
           for the student's input window.
           (Not used if just feedback is
           being created.)
  ,attrb - Field attributes used as
           defaults for all lines on this
           screen.  If none specified,
           (p,a) is assumed.  (Normal
           intensity, protected,
           alphanumeric.)
  ,att03 thru ,att24 - Attributes for any
           lines that differ from the
           default specified above.
  ,clear - Letter "y" (yes) if the
           previous display is to be
           cleared or "n" (no) if this
           text should overlay existing
           text.  (Creates either an "st"
           or "pr" statement.) Default =
           "y".  Ignored if just feedback
           is being created.
```

```
Macro name: aj
Keywords:
  ,ans1 thru ,ans9 - Up to nine equivalent
           predicted answers.
  ,op    - The operation code to be used
           (like "ca", "wa(1)", "un",
           etc.)
  ,row   - The number of the row to be
           checked for the student's
           input.  (Creates a "gt"
           statement.)
  ,attn  - The source of attention for
           this set of answers.  The
           default is the ENTER key.
           Other possibilities are PF keys
           or PA keys.
  ,count - Letter "y" (yes) if the correct
           or wrong-answer counter should
           be incremented, or "n" (no) if
           not.  Default = y
```

,group - A number indicating the instructional group into which the student should be placed for having typed this answer. (Used for branching purposes later.)

Following are the macro call statements needed under this system to create the same multiple-choice question shown earlier using a single macro.

```
mc tx
,title=Spanish Vocabulary
,t06=The Spanish word for "face" is:
,t08=   a) la cama
,t09=   b) la cara
,t10=   c) la casa
,row=19
gen
mc aj
,row=19,op=ca(1),ans1=b,ans2=&cara&
gen
mc tx
,t21=Right!
gen
mc aj
,op=wa(w),ans1=a c &cama& &casa&
gen
mc tx
,t21=Sorry, the answer is LA CARA.
gen
mc aj
,op=un
gen
mc tx
,t21=Please type one of these:  a  b  c
gen
```

Obviously, this system creates more work for the author in generating each question; however, the same two macros used here can also be used for text screens, all types of question screens, and even complex simulation screens. So, flexibility is the advantage of these single-function macros.

Macro coding tips - LCC and LCA statements

If you're doing any string manipulations or arithmetic calculations within your macro, then you're surely using local constant character (lcc) and logical constant arithmetic (lca) areas. (It helps to think of these "areas" as variables.)

Below is an example of an "lcc" variable used to create alternate answer statements. This code is taken from the "aj" macro described earlier. If the author specifies the ",op" keyword as "ca", then the alternate-answer op-codes need to be "cb's". Similarly, if the author specifies "wa", then the alternate-answer op-codes need to be "wb's". This code creates the proper alternate-answer statements:

```
if &ans2,n,.*+002       (If no alternate
                         answers are present,
                         then skip this entire
                         routine.)
set &c1,&op(1,1)        (Set the "lcc"
                         variable named
                         "&c1" to the first
                         character of the
                         author's ",op".)
&c1.b &ans2             (The op-code is
                         whatever character is
                         in "&c1" followed by
                         the letter "b". The
                         text field is whatever
                         the author has
                         specified in ",ans2".)
```

As an example of an "lca" variable, consider the following routine which checks to see whether the author is within the 6-character limit as a label is being created. The keyword "id" is being used to create the label. (This example also illustrates another very important technique -- the use of the "mn" op-code to send error messages to the author.)

```
if &id,n,.*+005         (If keyword ",id"
                         isn't being used,
                         then skip this
                         entire routine.)
set &a1,k'&id           (Set the "lca"
                         variable named
                         "&a1" to the
                         length of the
                         author's ",id".)
if &a1,lt,7,.*+002      (Skip the next 2
                         statements if the
                         length shown by
                         "&a1" is less than
                         7.)
@@mn ID IS TOO LONG     (Send a message to
                         the author
                         explaining the
                         invalid length.)
go .em                  (Stop generating
                         statements, and go
                         to the end of this
                         macro.)
lb &id                  (Continue by
                         generating the
                         label.)
```

It's helpful to define these "lcc" and "lca" variables in one location such as the very top of the macro, immediately following the keywords (the "ma" statements). Keeping them all in one place makes it easy for you to keep track of what ones you've used and avoids making

you scan the code to find them. You can also document them by using dummy statements that are never executed. You use "go" statements to prevent the dummy statements from being executed. Here's an example:

```
go .*+001
ty This var. is for length computations.
lca &a1
```

Documenting the generation date

By using a "cm" statement near the end of your macro, you can generate a permanent record of the date each time your macro is used. The author's course will contain a statement, immediately following the statements generated by your macro, which includes the calendar date. Here's the format of the statement that must be included in your macro:

```
cm 'mult' MACRO GENERATED ON &sysdat
```

When your macro is used, the variable "&sysdat" generates the calendar date.

Additional documentaton

Authors who use your macros may want a system for generating non-executable statements in their courses which merely document the course's logic. To satisfy this need, they may simply use macro call statements without a "gen" command. For example:

```
mc coment
,t01=This section is remediation for
,t01=students who miss more than four
,t01=questions.
```

The repetition of the same keyword ",t01" doesn't matter since the macro will never be generated. In fact, macro "coment" doesn't even have to exist.

A highly recommended procedure for developing IIS courses is to build the code in files such as TEXT files under TSO or SCRIPT files under VM. When all the code is ready to be generated, the author then signs on to IIS and issues a "get" command. (An even more efficient method is to do the sign-on and "get" in batch mode.) If you're using this method, then your authors may want to leave their comments and documentation in those files rather than occupying space in the course itself. To do this, they can simply begin any line of their file with an asterisk. Following is an example:

```
.
.
,t20=Press ENTER, please.
gen
* * * * * * * * * * * * * * * * * * * * * * *
* This section is remediation for
* students who miss more than four
* questions.
* * * * * * * * * * * * * * * * * * * * * * *
mc text
,t01=Review
.
.
```

When the "get" command is issued from IIS author mode, the asterisked lines will be ignored and no Coursewriter statements will be generated from them.

Looping or Branching among macros

If you're very careful, you can save some space in your macro by creating a loop. Your macro can contain labels beginning with a period, and you can branch to those labels multiple times while statements are being generated. You can even branch to a label located in another macro, which would actually be a subroutine since it's usable by multiple macros. The following simple macro contains one such subroutine labelled ".sub":

```
mc tx
ma &t03=,&t04=,&t05=,&t06=,&t07=,&title=
pt fld001(p,a)&title
lb .sub
pt fld003(p,a)&t03
pt fld004(p,a)&t04
pt fld005(p,a)&t05
pt fld006(p,a)&t06
pt fld007(p,a)&t07
if &rtn,eq,tf,.rtf
if &rtn,eq,mc,.rmc
em
```

The variable "&rtn" is used to determine the return label after the subroutine is used. Other macros can use this subroutine by first loading "&rtn" with the appropriate return label and then simply branching to label ".sub". Following is an example:

```
.
.
set &rtn,rtf
go .sub
lb .rtf
.
.
```

Summary

Some suggestions for designing, coding, and documenting IIS Coursewriter macros have been presented here. Hopefully you, the experienced Coursewriter author, have found a technique or two that will be of value to you.

IMPLEMENTATION OF A SOFTWARE INTERFACE WITH BAR CODE DATA ENTRY FOR AN INTELLIGENT INTERACTIVE VIDEODISC INSTRUCTIONAL SYSTEM

Gregory R. Bauer - Indiana University at South Bend

Albert E. Miller and Eugene W. Henry - University of Notre Dame

ABSTRACT

This paper describes a Bar Code/Interactive Videodisc (BC/IV) System that utilizes bar code data entry to control an IBM PC and a Sony LDP-1000 videodisc player. User control is accomplished through a system control program which linkes the hardware components of the system together and, in turn, links the hardware to software applications modules related to the content of the videodisc. Details of the linking process are given in the paper.

The paper also describes applications of the system in contexts such as computer-assisted instruction and computer-based training along with other application areas. It gives an overview of bar code and interactive videodisc technologies including the current status of each.

Finally, the paper states two specific conclusions about the BC/IV System. The system demonstrates that bar code data entry is a viable alternative data entry mode compared to a keyboard. It also demonstrates that bar code data entry can be used to control both a microcomputer and a videodisc player.

INTRODUCTION

Technological advances in the past decade have caused a dramatic change in the way the world operates. The development of the microprocessor and its delivery system, the microcomputer, have played a major role in these changes.

Two other technologies have begun to show promise of major impact in many facets of everyday activity. Bar code technology and videodisc technology are both flourishing today because of the microprocessor.

Since the introduction of the microcomputer, the primary form of data entry has been the keyboard. However, the keyboard has been less than enthusiastically used by many people because it is felt that one must be an excellent typist to effectively use it. This feeling of need for typing skills is brought about by the fact that the microcomputer keyboard is exactly the same, with few exceptions, as the traditional typewriter keyboard. In fact, many people have been reluctant to use a microcomputer because of a fear that "hunt and peck" will no longer be adequate.

Many people have wondered about the possibilities of alternative modes of data entry to the keyboard. Efforts to search for alternatives have not received much positive attention.

However, the number of people using microcomputers has increased dramatically in the past several years. The result is that the interest in alternative data entry technique has been received with new attention and emphasis by researchers. It is clearly evident that alternative modes of data entry are being studied.

One of the alternative forms of data entry that has proven viable in certain situations is bar code. Bar code is used for identification purposes and in that context has proven to be highly successful as a data entry mode in material handling situations.

Since bar code has proven to be a fast, accurate and easy-to-use method of data entry to a microcomputer, it should be considered for data entry in situations other than materials handling. The possibility bears exploration.

It has already been clearly demonstrated by a number of researchers that a microcomputer using a keyboard can be effectively used to control a videodisc

player. Could bar code be a viable alternative to the keyboard in controlling a videodisc player?

The purpose of this paper is to describe work that has been done in the College of Engineering at the University of Notre Dame to find answers to the aforementioned questions. The work has resulted in the development of the Bar Code/Interactive Videodisc (BC/IV) System. The BC/IV System uses bar code for data entry including control of an intelligent videodisc player.

The paper is broken down into four major sections. The first section contains a brief overview of both bar code and interactive videodisc along with a description of current activities in both technologies. The second section describes the hardware and software components of the BC/IV System and how these components are integrated to form the system. The third section describes some applications for the system. Concluding remarks are made in the fourth section.

BAR CODE AND INTERACTIVE VIDEODISC TECHNOLOGIES: AN OVERVIEW

Advances in technology have had, and will continue to have, a tremendous impact on the delivery of instruction at all levels. Leading the way has been the increased popularity of the micrcomputer.

This popularity has brought the power of computers both education and training. With it has come the development of complimentary technologies which have given rise to new and innovative applications in both areas.

Two of those complimentary technologies are bar code and interactive videodisc. Both technologies have many applications in their own right but the number of their applications is increased because both technologies are compatible with the microcomputer.

The purpose of this section is to describe the basis and current state of affairs in both bar code and interactive videodisc technology. Even as this description is being written, advances in both technologies are being realized and new applications are being developed.

Bar Code and Bar Code Systems

Most people have seen bar code. It looks like a patch of stripes usually black or blue on a background of white. Bar code is usually found on every product purchased in a retail store. Its principal function is for purposes of identification and inventory control.

As the name implies, bar code is a representation of data in a symbolic form. This representation is made up of bars and spaces printed or embossed on a surface. An example of bar code is shown in Figure 1.

Figure 1. Example of a Bar Code Symbol

These bars and spaces are arranged in a predetermined pattern that follows clearly defined rules. The pattern for one character differs from the pattern of another character by the use of different numbers and different widths of bars and spaces. The height of the bars is not, in most cases, important.

The original purpose of bar code was identification. The particular pattern attached to some entity identified that entity as being unique. This is somewhat analogous to the use of a fingerprint or social security number. Bar codes are most often used for inventory management or for product pricing. The latter use is perhaps more familiar to most people and works in the following way:

An item in a retail store is brought to the checkout counter. The bar code symbol on the item is read by a bar code reader which identifies the item. The information about the product's identity is then passed to a unit which matches the identity with a predetermined price for the item. The price is then registered on a display for the purchases and is added to a commulative total. The process continues until all items to be purchased by the customer have been processed.

The primary advantage of bar code and its associated reader (scanner) in the example just presented is the speed and accuracy at which the code can be read and processed. Human error is greatly reduced which results in greater productivity. Current bar code technology also allows data represented by a bar code symbol to be read and stored in a computer where further processing may occur. Thus, the use of bar code gives an alternative mode of data entry to the keyboard.

The Nature of Bar Code: What is the process involved in taking a bar code symbol such as the one in Figure 1 and converting the data into data that can be understood by the computer? First of all, reading a bar code is simply a process of distinguishing between dark and light areas (bars and spaces) and comparing their relative widths. This process requires the use of a scanning device that can illuminate the bar code symbol and

interpret the reflected light.

Figure 2 shows a cross-section of a commonly-used scanning device known as a wand. Light that is generated within the wand, is projected through the tip, and then reflected back through an aperture to a light detector. To properly scan a bar code symbol, the dark areas (bars) must reflect as little light as possible to the light detector. It is therefore necessary that the aperture opening be no larger than the width of the narrowest bar.

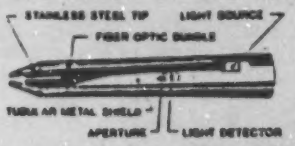


Figure 2. Cross-section of a Bar Code Wand

Normally, the data collected by the wand through the light detector is immediately passed to a decoder which converts the data to a set of binary zeroes and ones (ASCII Code) that can be understood by the computer. Some wands are capable of storing the data (up to as much as 4000 characters) collected for later passage to the computer.

As one can imagine, there are a wide variety of bar codes available for use. The American National Standards Institute (ANSI) has developed a standard for bar code, but this standard is sufficiently flexible so that many different bar codes can exist.

Another organization, Automatic Identification Manufacturers (AIM), a division of the Materials Handling Institute, is also involved. It has published a Uniform Symbol Description (USD) for each of eight bar codes. Most people who use bar code recognize these eight codes as somewhat "official" but they are not obligated to use one or more of the eight codes. However, manufacturers of bar code scanners and decoders have tended to use the USD in the manufacture of their equipment so that USD have become the "defacto" standards.

A notable exception to the eight USDs mentioned above is the Uniform Product Code (UPC) which was developed by the grocery industry and is used by many other retailers. For whatever reason, UPC is not one of the eight AIM bar codes.

Bar codes are usually given a name that describes in some fashion the nature of the code pattern. For example, Code 11 is so named because it consists of eleven characters. On the other hand, Code 3 of 9 (also known as Code 39) is so named because it uses a pattern of nine elements (bars and spaces) with three of the elements being "wide".

Table 1 gives a summary of the characteristics of the seven most widely used bar codes. In addition, corporate sponsors (scanner and decoder manufacturers) are also listed. The importance of the characteristics shown in the table should be self-evident. Alphanumeric and variable length codes have greater flexibility for the user. This flexibility is also reflected in the number of possible data characteris available. There is a 128-character upper limit which reflects the number of possible American Standard Code for Information Interchange (ASCII) characters. This is important when using a computer. The data security and self-checking characteristics are important, depending upon the code application.

A Specific Example - Code 39. Code 39 is of interest because it allows the use of both numeric and alphabetic characters. A list of the characters and their bar code representation is shown in Figure 3. Note that the binary representation is not the ASCII code. The conversion to ASCII code is done by the decoder.

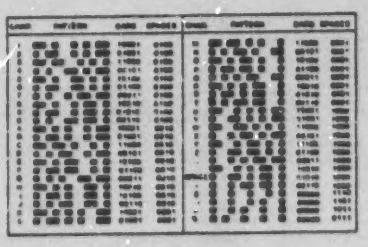


Figure 3. Table of Code 39 Symbols.

A "1" is used to represent a wide bar or space and "0" is used to represent a narrow bar or space.

The name Code 39 is derived from a code structure that has three wide elements (bars and spaces) out of nine elements. The nine elements representing a data character consist of five bars and four spaces.

A typical symbol in Code 39 may consist of a single data character or a string of data characters. In either case the symbol must be preceded and followed by a start/stop character. Preceding and following the start/stop character is a "quiet zone" which is simply a blank space. Figure 4 shows an example of a Code 39 symbol and how it is converted into it binary representation.

Two additional characteristics of the code should be mentioned. The ratio of the width of a wide bar or space to the narrow bar or space, ranges between 2:1 and 3:1. The space between data charac-

Table 1

Summary of Common Bar Codes and their Characteristics

Code	Main Field of Use	Alpha-Numeric	Variable Length	Self-Checking	No. of Data Chars.	Data Security	Corporate Sponsor
2 of 5 Inter-leaved	Industry	No	Yes	Yes	10	High	C Identics
Code 39	Industry	Yes	Yes	Yes	43-128	High	Intermec
Codabar	Medical Photo-graphy	No	Yes	Yes	16	High	Monarch
2 of 5 Datalogic	Industry	No	Yes	No	10	Moderate	
UPC	Retailing	No	No	No	10	High	--
Code 126	New	Yes	Yes	Yes	103-128	Moderate	C Identics
Code 93	New	Yes	Yes	Yes	47-128	High	Intermec

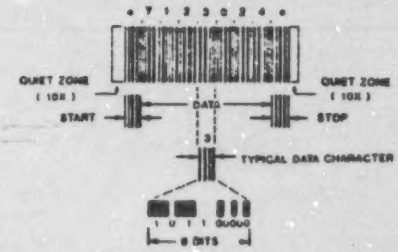


Figure 4. Anatomy of Code 39 Symbol

ters (such as between 7 and 1 in Figure 4) ranges from the width of the narrow bar or space to nine times that width. The space between characters is known as the inter-character gap.

Finally, the number of data characters in a Code 39 symbol is limited only by the scanner used to read the symbol. Most scanners have a range of one to thirty data characters per single read.

Bar Code Today. The current primary use of bar code continues to be for identification purposes. Labels containing a bar code symbol are easily and accurately read with both portable and fixed scanners.

As more and more interfaces are being developed between bar code and the microcomputer, some people are beginning to consider bar code as an alternative data entry mode to the keyboard. While little is evident in the literature about this use of bar code, it is clear that this possibility will be explored more fully in the near future.

Interactive Videodisc.

Unlike bar code, interactive videodisc is not familiar to most people. Its primary use is in educational and training applications and is considered by many people to be one of the most important advancements along with the microcomputer, in the field of applied learning technology.

The Nature of Interactive Videodisc. In order to understand the nature of interactive videodisc, it is necessary to break the term down into its two component parts. One component (videodisc) is dependent upon hardware; the other component (interactive) is dependent upon software.

The videodisc itself exists because hardware was developed which enabled the recording of an analog video signal onto a metal surface covered with a smooth plastic coating. The recording technique permitted the recording of both still frame and motion sequences on the same disc. Along with video, sound could also be recordable on the disc.

The recording technique also had an additional advantage. A great deal of data could be recorded on a very small amount of surface. The storage capacity of the videodisc is much greater than

that of ordinary videotape. In addition, the quality of the analog signal does not diminish over a period of time as is the case with videotape.

The "interactive" component of the term is software-dependent. To be "interactive", the videodisc must be under the control of the person who is accessing or using the videodisc. Random access at any time to any frame on the videodisc is the goal. To do so, the user must have at his disposal at least a microprocessor. By controlling the microprocessor through a software interface, the user is able to directly control the videodisc and achieve the goal of random access.

The Interactive Videodisc System. In order to use an interactive videodisc, a user must have access to a videodisc system. The videodisc system has five basic components: 1) a videodisc player, 2) a computer controller, 3) input devices, 4) output device(s) and 5) software to drive the system.

The most popular type of videodisc player is one that uses a laser to read data from the videodisc. The advantage of this type of player is that there is no physical contact between the laser and the videodisc. Hence, the videodisc surface is not worn because of contact. Other types of videodysic players use a stylus to read the data from the videodisc but these players can eventually cause wear to the videodisc with potential loss of video signal quality and even loss of data.

Many videodisc players are known as "intelligent" because they have a built-in programmable microprocessor which controls the playing of the videodisc. These players may be operated manually by the user with a remote control unit or remotely through the use of a microcomputer. In both cases, forward/reverse play from one point (frame) to any other frame on the videodisc is possible.

The second component of the interactive videodisc system is the computer controller. The controller is usually a microcomputer. The controller is programmed with an applications program, written in some high-level language and allows the user to interact with the data on the videodisc in some organized fashion. The computer controller must work in partnership with the microprocessor on the videodisc player; this partnership is controlled by the applications program.

In order to be useful, an interactive videodisc system must have a way to input data and com[illegible]rom the user. The traditiona[illegible]nput has been the keyboard bu[illegible]e modes of input are now bei[illegible]ong others, these alternative [illegible] include bar code and bar code recorders, keypads, tablets, touch sensitive screens and voice recognition.

The fourth component of an interactive videodisc system is the output device. This device must be capable of displaying a standard video signal such as the one from the videodisc. At the same time, it is desirable for the output device to be able to display screens generated by the applications program on the same screen at the same time as the image from the videodisc. To achieve this duality of images on the output device, the microcomputer must have hardware and software which will permit the desired end result. This result of a dual image on the output device is usually referred to as "graphics overlay" and found more and more in current interactive videodisc systems.

Finally, the interactive videodisc system must have software to drive the system for the user. The applications program should be able to take advantage of the interactive potential of the computer controller and the potential for a wide variety of media presentation formats available through the videodisc. Of course, the applications program must also have some useful purpose or objective(s) for the user to benefit from the use.

Interactive Videodisc Today. Much attention is currently being focused on the use of interactive videodisc, especially in the fields of education and training. Its use offers the interactivity and power of the computer and the multi-media capabilities of the videodisc. Many educators and trainers agree that the interactive videodisc is a variable tool to accomplish their desired objectives.

However there are several factors which are limiting the current acceptance of the interactive videodisc. Among others, these factors include: 1) cost, 2) lack of good applications programs and 3) lack of erasability of information on the videodisc itself.

The cost of purchase of and production on an interactive videodisc system is a legitimate concern. While the cost of hardware for the system is decreasing and will continue to do so, many people are not yet convinced of its cost effectiveness. It is likely, however, that the hardware cost will become much more affordable, especially in light of an increase in networking capabilities.

The more serious problem in terms of cost is the production of the information recorded on the videodisc and the applications program needed to control the system. The cost here is essentially human, there is general agreement that it takes a team of five to seven people to

produce a quality videodisc and applications program. Human costs tend to increase, not decrease.

The art and science of designing a quality applications program is not easy to master. Because of its random access capabilities, the videodisc offers great opportunity for branching capabilities. Unfortunately, most application program designers tend to think in a linear fashion and it has been difficult for these designers to "switch gears". This problem will be alleviated with time.

A serious limitation of current videodisc technology is that, once data is recorded on the videodisc, it cannot be changed. Old data cannot be erased or new data recorded over the old. Thus, if data on a videodisc needs to be updated, a new videodisc must be produced, resulting in additional cost. This problem is on the verge of solution and even at this writing, it is likely that at least one manufacturer has perfected an erasable videodisc.

Even with the above mentioned limitations, the future of interactive videodisc appears bright. Many people are working on solutions to the problems, none of which seem insolvable.

THE BAR CODE/INTERACTIVE VIDEODISC SYSTEM

The Bar Code/Interactive Videodisc (BC/IV) System is an example of the interactive videodisc system described in the previous section. The purpose of this section is to describe the hardware and software components of the system and how the system integrates these components.

System Components

As with any system, it takes both hardware and software components to make the system operate effectively. This section will include a discussion of the specific components of the BC/IV System.

Hardware. As mentioned previously, the BC/IV System hardware is modeled after the interactive videodisc system described earlier. First, the videodisc player is a Sony LDP-1000. This particular player is an example of an "intelligent" videodisc player because it has a Z-80 microprocessor built into it. The player may be controlled from either a hand-held control unit or from a microcomputer. It plays a 12" videodisc using a laser. Because of the size of the videodisc, there are approximately 54,000 frames on which data can be stored on each side of the videodisc.

The computer controller for the BC/IV System is an IBM PC with 512 kilobytes of random-access memory. The IBM PC is configured with a dual, double-sided disk drive. Internally, the IBM PC is controlled with an 8088 microprocessor. The PC has an asynchronous communications board with a serial port for the interface to the bar code reader. It also has a multifunction board which provides the interface to both the Sony LDP-1000 videodisc player through a serial port and the interface to the Epson FX-80 dot matrix printer through a parallel port. Finally, there is a color graphics display board which provides the interface to a Sony or an AMDEK monitor.

One of the unique features of the BC-IV System is the input device for the user to communicate with the system. Instead of the keyboard, all input from the user is entered using bar code. The bar code for a given input is read using an Intermac Model 9800 bar code reader with a portable, visible-light wand. The reader is set to read a Code 39 symbol. After reading the symbol via the wand, the reader translates the signal received into the corresponding American Standard Code for Information Interchange (ASCII) format, appends an ASCII carriage return symbol and sends the ASCII characters to the IBM-PC where the data is then read and processed by the system control program.

The BC/IV System utilizes three output devices. It displays PC System output to either a Sony or an AMDEK monitor. The Epson FX-80 dot matrix printer is used to print output in the form of bar code. The third output device is a Sony composite video monitor which displays images from the videodisc.

Software. An interactive videodisc system must have software to enable the hardware to perform. The BC/IV System is no exception.

The software component of the BC/IV System consists of two basic programs. The major program is the system control program which controls the IBM PC and the Sony LDP-1000 videodisc player. The second program is one that enables the dot matrix printer to generate output in the form of bar code symbols.

The system control program is written in IBM Advanced BASIC which is a product of Microsoft. Its purpose is to allow the user to access the information contained on the videodisc in a meaningful fashion. The program utilizes the communication ports of the PC to receive user input from the bar code reader and to send control commands to the videodisc player. The system control program has the difficult task of coordinating the input to and the output from the PC so that control signas do not interfere with each

other.

For the user, the system control program utilizes a menu and prompt approach. The user is continuously shown a screen of instructions or a menu screen and is always given a prompt for the data to be input to the system.

The supporting software program which prints output as bar code is a product of Dataflow Technologies Incorporated. The software allows a user to input either a single or a string of alphanumeric characters to the software program and then proceeds to output the characters in one of several different bar code formats. For the BC/IV System, code 39 is used with the Dataflow program.

System Interpretation of Hardware and Software.

In order to understand how the BC/IV System integrates its hardware and software components, a flowchart is shown in Figure 5. The flowchart shows how data is input and acted upon by the BC/IV System. The figure also shows that the System Control Program (SCP) is in complete control of the BC/IV System. It receives data from the user through bar code input, sends appropriate data to the videodisc player or processes the data itself and then waits for the next user input.

APPLICATIONS OF THE SYSTEM

It is one thing to create a system. However, the creation of the system has little value unless the system has some useful applications. The purpose of this section is to describe some general applications for the BC/IV System and to describe a specific example of the system control program for a selected videodisc.

General System Uses

The primary use of the BC/IV System is to integrate its selected hardware and software components for the expressed purpose of delivering various educational and training applications to a single user or a group of users. Thus, the effectiveness of the BC/IV System must be judged on two criteria: 1) the ease of system operation for the user and 2) the quality of the educational and training applications delivered by the system.

One general use of the BC/IV System is to deliver computer-assisted-instruction (CAI). Many educators are recognizing the potential value of CAI as an effective mode of learning for students at nearly every educational level. Since the

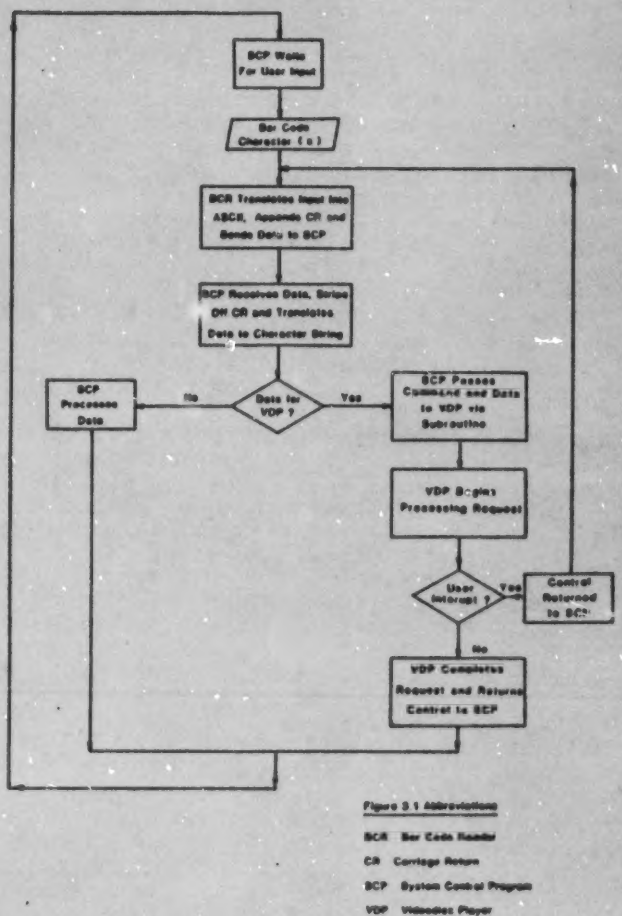

SCP Waits For User Input
Bar Code Character (s)
BCR Translates Input Into ASCII, Appends CR and Sends Data to SCP
SCP Receives Data, Strips Off CR and Translates Data to Character String
SCP Processes Data
No
Data for VDP ?
Yes
SCP Passes Command and Data to VDP via Subroutine
VDP Begins Processing Request
User Interrupt ?
Yes
Control Returned to SCP
No
VDP Completes Request and Returns Control to SCP
Figure 5.1 Abbreviations
BCR Bar Code Reader
CR Carriage Return
SCP System Control Program
VDP Videodisc Player

Figure 5. Flowchart of System Integration of Hardware and Software.

BC/IV System has the capability of being highly interactive with the user, it provides excellent delivery capabilities. In addition, the interactive videodisc component gives a multimedia capability that moves the presentation of the content of the application being presented much more enjoyable and meaningful to the user.

Related to CAI is computer-based-training (CBT) and the BC/IV System is equally useful in delivering this type of application. There is an indistinguishable difference between CAI and CBT but suffice it to say that CBT applications often present content relating to a procedure. At any rate, CBT application presentations parallel the CAI application presentations.

A second use of the BC/IV System is in a testing situation. No content is presented as with a CAI or CBT application. A user is given a series of test

questions to which a response is made using the bar code wand. The answers can then be saved in a sequential data file and printed out on the dot matrix printer in bar code. An obvious advantage of printing student responses in bar code to test questions is security; it is literally impossible to sight-read a bar code symbol.

A third area in which the BC/IV System might be used is in a large group instructional setting such as in an introductory engineering course. Each student has a bar code wand and a set of bar code symbols at his seat. The instructor asks the class a question related to the material being presented. Each student responds by wanding a bar code response. These responses are immediately processed and summarized for the instructor on a panel at the rostrum. Based upon the summary, the instructor has immediate feedback on whether or not the material presented was understood by the students.

A fourth use of the BC/IV System is in a word or text processing situation. By linking with a word processing package, the user enters text with the bar code wand instead of the keyboard. Because of the nature and uses of bar code data entry, it is likely that in some situations, a user may prefer using the bar code wand instead of the keyboard.

A Specific System Example

In order to demonstrate that the BC/IV System is operational, a specific system control program for a specific videodisc has been developed. This application was designed for use by a single user and demonstrates the capability of the BC/IV System.

The videodisc chosen for demonstration is one produced by Video Vision. The videodisc concerns both still photographs and video motion sequences taken during the flight of STS-5, one of the NASA's space shuttle missions.

The application has four basic modules. Two modules are related directly to the videodisc content; one module accesses the still photographs and the other module accesses the six video motion sequences. While the pictures accessed by both modules are on the videodisc, their access is controlled by the system control program, and, utimately, by the user.

In addition to the two aforementioned modules, the user has access to two other modules. One module is a glossary of abbreviations and acronyms. The user can access this module at literally any time or at any desired point. This module contains definitions and acronyms which were provided in documentation which accompanies the videodisc. However, the organization of the screens presenting the content of the glossary and access to those screens are a part of the system control program.

The fourth module determines what the user has learned from exposure to the content of the videodisc. Multiple-choice questions are presented to the user who responds by entering a bar code symbol. The user has the opportunity to change the response to any (or all) questions as often as directed. However, the system control program does not permit the user to return to other modules to access information requested by the question.

After the user is satisfied with the choice responses, the responses are saved in a sequential data file which has been named by the user. Then, or at some later time, the user's data file can be accessed and the responses to the questions can be printed in bar code symbols.

CONCLUSION

The creation and implementation of the BC/IV System demonstrates that the use of bar code input is a viable alternative to the use of a keyboard. It is possible to create a bar code symbol for every keyboard character or function. The speed and accuracy of bar code in data entry gives it any advantages over the keyboard.

Another important fact demonstrated by the system is that a satisfactory interface can be made between a microcomputer and videodisc player. This interface, along with both devices, can be satisfactorily controlled by bar code input.

REFERENCES

1. American national standard for materials handling-bar code symbols on unit loads and transport packages, New York, N.Y., American National Standards Institute, 1983.
2. Baur, G.R., The use of bar code input/output in an intelligent interactive videodisc instructional system. MSEE Thesis, University of Notre Dame, August 1984.
3. Baur, G.R., Miller, A.E. and Henry,E.W. Bar code input/output in an intelligent interactive videodisc system, reviewed paper presented at the Second Annual IEEE Edu. Comp.Con., San Jose, California, November 1984.
4. DeBloois, Michael, ed. Videodisc/Microcomputer Courseware Design, Englewood Cliffs, N.J., Educational Technology Publications, 1982.

5. Pigford, V.D., Interactive videodisc: a model for authoring systems and a software interface for microcomputers, MSEE Thesis, University of Notre Dame, August 1984.
6. Uniform symbol description - 3: Code 39, American Identification Manufacturers (a division of the Materials Handling Institute), 1982.

REPORT OF A PILOT STUDY TO EXAMINE THE USE OF INTERACTIVE VIDEODISC CARDIOPULMONARY RESUSCITATION INSTRUCTION

MARGARET J. A. EDWARDS, R.N., M.A., UNIVERSITY OF CALGARY

KATHRYN J. HANNAH, R.N., PH.D., UNIVERSITY OF CALGARY

ABSTRACT

The purpose of the study was to examine the efficacy of the American Heart Association interactive videodisc system for community cardiopulmonary resuscitation (CPR) instruction.

In classical experimental tradition, an experimental group was taught CPR through the interactive videodisc system while a control group was taught in the traditional method by instruction. All groups were retested at 3 months. Data collection instruments included a demographic data form, a cognitive knowledge pre-test, post-test and retention test, and a skills performance test.

The findings of the pilot study indicate that the interactive videodisc system is at least as effective as traditional CPR instruction in the training of lay persons in this skill. The cost effectiveness of the interactive videodisc was found to vary according to the numbers of persons to be trained per time period.

Should further study substantiate these results, use of such a system would begin to address issues such as variability of instructor standards and delivery of CPR instruction to people with limited access to traditional courses.

* * * * *

The linkage of computer technology with interactive videodisc systems has allowed for the development of innovative methods of instruction. The American Heart Association has developed a self-training system in cardiopulmonary resuscitation (CPR) that utilizes both technologies. The CPR learning system is composed of a Sony VDP-1000 laser optical videodisc player, two monitors, an Apple II computer, a random access audio machine, a light pen, and a sensorized manikin. Although the system is being marketed, there has been only limited internal testing of the components.

The purpose of this study was to begin to marshall evidence as to the efficacy of the American Heart Association interactive videodisc system as a strategy for community CPR instruction. Four research questions were used to guide the investigation.

METHOD

Population

The population for the Pilot Study included sixty-five (65) employees of a major company. The company routinely offers CPR certification courses to employees. Those people who enrolled in the Heart Saver course (single person CPR and conscious obstructed airway) self-selected into control (16) and experimental groups (27). Those people who enrolled in the Basic Life Support course (single and two person CPR, conscious and unconscious obstructed airway, and infant CPR) were assigned to either the control group (6) or experimental group (16) based on their accessibility to the CPR learning system.

Experimental Design (Figure 1.1)

	Control Group	Experimental Group
Pretest	Written	Written
Treatment	Instructor Taught	Interactive Videodisc CPR Instruction
Post-Test	Written	Written
	Skills Performance	Skills Performance
Retention Test (3 months)	Written	Written
	Skills Performance	Skills Performance

Pre-test: A written cognitive knowledge test was administered to all subjects prior to the commencement of CPR instruction. Different tests were used for Heart Saver courses and BLS courses.

Treatment: The control group in each pilot study was taught by AHF certified instructors following the traditional format. The traditional format is taken to mean a lecture utilizing slides, films, demonstrations and manikin practice. In the Pilot Study, the same two instructors taught the entire control group. The use of a limited number of instructors to teach the control group was deemed to limit the variability due to differences in instructional methods, presentations and standards.

The experimental group was taught CPR through the use of Actronics Inc./American Heart Association CPR Learning System previously described. Two subjects used the CPR Learning System at a time, in four (4) hour blocks. A research assistant was present in an anteroom throughout the use of the CPR learning system to monitor the time taken by each pair of subjects and to provide technical assistance.

Post-test: Following completion of instruction, in any group, the same (pre-test) written cognitive knowledge test was administered to all subjects.

To determine performance skills, each subject was asked to demonstrate the skills required by each course (Heart Saver or BLS). Those subjects in the control group were evaluated by two independent Canadian Heart Foundation (CHF) instructor trainers using the very specific CHF guidelines. The CPR learning system evaluated the skills performance of those in the experimental group using the same CHF standards. The CPR skills performance of those in the control group were recorded on a thermoengraved paper tape produced by a recording manikin. The skills performance of the experimental group was analysed by and recorded on computer files.

Retention Test: All subjects who passed initially, in all groups, were asked to report to the CPR project room at a specified time. When the person arrived, they were asked to perform a one-person rescue including the initial sequence and one (1) minute of CPR. Both the initial sequence and the skills performance were recorded as in the initial performance test. Following the skills test, the same (pre-test, post-test) written cognitive knowledge test was administered.

Instruments

All participants were asked to complete a demographic data form in addition to the cognitive knowledge pre-test and post-test and skills performance test.

FINDINGS

The findings of the pilot study will be reported as responses to the questions posed initially. In addition to addressing the research questions, a brief discussion of supplementary findings will be presented.

1. What is the nature of the relationship between group membership (control/experimental) and skill performance score?

Table 1.0 Frequency distribution of skill performance by group and type of course (n=62)

Skill Performance Category	Total Group (%)	Heartsaver Control (%)	Heartsaver Experimental (%)	BCLS Control (%)	BCLS Experimental (%)
PASS	51(82.3)	17(100)	16(64)	5(100)	13(86.7)
FAIL	11(17.7)	0	9(36)	0	2(13.3)

Table 1.1 Fisher's Exact Test ($n < 21$) or Yates' Corrected χ^2 Statistic ($n > 21$) for skill performance

	$c\chi^2$	df	probability
Total Group (n=62)	4.23917	1	0.0395*
Heartsaver (n=42)	5.45753	1	0.0195*
BCLS (n=20)			0.89474

*significant at $p = 0.05$

Table 1.0 presents the frequency distributions for skill performance by group. The next table, Table 1.1 illustrates the result of subjecting the variable by group to a Crosstabs analysis. Where the $n > 21$, a Fisher exact probability is reported, and where $n < 21$, Yates' corrected chi square and its probability is reported. There at first appears to be a significant difference in skills performance between the experimental and control groups at the Heart Saver level. This was, in fact, the case. However, all eleven failures in the experimental group were due to "jabbing" (unequal compression and relaxation cycles). The version of the software used by the experimental group was found to never mention 'jabs' prior to recording them on a subject's results. Also, experimental subjects had to determine, on their own, whether or not their performance was adequate to be recorded as their examination. The control group, however, was coached to mastery. This would seem to account for the difference in skills performance. We have just completed another pilot study using a new software release. The new software has a remedial learning sequence for jabs and also tells the trainee whether or not the skills performance meets the standards. In this way, the trainee is then able to record only an acceptable performance, as is done in traditional CPR instruction. In our latest study, all experimental group trainees were able to produce passing performances after their initial instruction. Many other crosstabulations between skill and demographic variables were completed. With the exception of skill by group, no other crosstabulations yielded significant results. Therefore, based on the new version of the software, it can be said that there is no relationship between group membership and skills performance.

2. What is the nature of the relationship between group membership and gain score on the cognitive knowledge test?

Table 2.0 Frequency distribution of categorized gain scores by group and type of course

Gain Score Category	Total Group (%)	Heartsaver Control (%)	Heartsaver Experimental (%)	BCLS Control (%)	BCLS Experimental (%)
-15 to 0	10(16.1)	1(5.9)	1(4.0)	2(40)	6(40)
1 to 15	18(29.0)	5(29.4)	8(32.0)	2(40)	3(20)
16 to 29	19(30.6)	6(35.3)	7(28.0)	1(20)	5(33.3)
30 to 55	15(24.2)	5(29.4)	9(36.0)	0	1(6.7)

Table 2.1 T-test statistics for raw gain score

	t-value	df	2 tailed probability
Total Group (n=62)	-0.41	60	0.68
Heartsaver (n=42)	-0.52	40	0.607
BCLS (n=20)	1.0	18	.331

Table 2.0 shows the frequencies of the categorized gain score by group. People who received 85% on the post-test were considered to have passed. Table 2.1 illustrates the results of subjecting the means of the raw gain scores to t-test analysis. In this question, it can be seen that there is no significant difference between groups. This is a finding that was anticipated as well as desired. These results indicate that the computer system was at least as effective as traditional instruction in teaching the cognitive portion of CPR.

Again, many crosstabulations examining the cognitive knowledge gain score in relation to the demographic variables yielded nothing of significance.

It can be concluded from the results that in the pilot study there is no relationship between group membership and cognitive knowledge gain score.

3. What is the nature of the relationship between group membership and cognitive knowledge retention at 3 months?

Table 3.0 Frequency distribution of categorized cognitive knowledge retention (3 mos.) scores by group and type of course (n=34)

Retention Knowledge Score Category	Total Group (%)	Heartsaver Control (%)	Heartsaver Experimental (%)	BCLS Control (%)	BCLS Experimental (%)
71% - 80%	4(11.8)	1(7.7)	1(8.3)	0	2(28.6)
81% - 90%	14(41.2)	4(30.8)	6(50.0)	2(100)	2(28.6)
91% - 100%	16(47.1)	8(61.5)	5(41.7)	0	3(42.9)

Table 3.1 T-test statistics for raw cognitive knowledge retention scores

	t-value	df	2 tailed probability
Total Group (n=34)	-0.27	32	0.786
Heartsaver (n=25)	0.05	23	0.959
BCLS (n=9)	0.59	7	.575

Table 3.0 represents the frequency distribution of categorized cognitive knowledge retention scores by group and type of course. The lowest percentage received was 75%, while two subjects achieved 100%.

While the purpose of our study was not to examine retention per se, some consideration was given to examining cognitive knowledge retention. Using students' t-test for paired samples, the decrement of cognitive knowledge was not significant at p=0.678.

Table 3.1 indicates that, while cognitive knowledge decreased significantly, there is no relationship between group membership and cognitive knowledge retention scores. The t-tests do not yield probabilities that would even suggest trends.

Crosstabulations were performed to examine the relationship of the cognitive retention score with the demographic variables. Again, no significant relationships were found.

From these results, then, it can be said that, although cognitive knowledge decreased, there is no relationship between group membership and cognitive knowledge retention score.

4. What is the nature of the relationship between group membership and retention skill performance at 3 months?

Table 4.0 Frequency distribution of retention skill performance by group and type of course (n=34)

Retention Skill Performance Category	Total Group (%)	Heartsaver Control (%)	Heartsaver Experimental (%)	BCLS Control (%)	BCLS Experimental (%)
PASS	10(29.4)	2(15.4)	4(33.3)	1(50)	3(42.9)
FAIL	24(70.6)	11(84.6)	8(66.7)	1(50)	4(56.1)

Table 4.1 Fisher's Exact Test ($n < 21$) or Yates' Corrected χ^2 Test Statistic ($n > 21$) for retention skill performance

	c χ^2	df	probability
Total Group (n=34)	.47769	1	0.4895
Heartsaver (n=25)	.14824	1	0.7002
BCLS (n=9)			0.49742

Table 4 indicates the frequency of retention score performance by group and type of course. Again, though not focussing on retention, some comments must be made about retention skill performance. Other studies cite retention skill passing performance rates anywhere from 13% to 83%. It would seem that our results then certainly fall within these broad bounds. The most frequently cited 3 month skill retention rate is 30%. It is apparent that the total retention in this pilot study, at 29.4% is very comparable.

The rate difference between the control group and the experimental group in the Heart Saver course is of particular interest. However, as Table 3.1 indicates, this difference is not statistically significant. The results noted in this table further indicate that no significant difference is evident in skills retention performance between the groups.

For the pilot study, we chose to only evaluate skills performance as to pass or fail, therefore, we are unable to pinpoint precisely which

skills deteriorated. Simple scanning of the performances indicates that most people were able to correctly perform the initial assessment sequence with the exception being the pulse check. The major problems in compressions and ventilations were jabs, shallow compressions and inadequate ventilation. Only 1 potentially injurious performance was noted.

No subjects in either group reported experience with or further instruction in CPR since their initial training.

It would seem from these results that, although a decrease in skill performance is noted at 3 months, there is no relationship between that decrement and group membership.

SUPPLEMENTARY FINDINGS

One of the major questions asked of us is "But is it cost effective?" Again, although this question was not the focus of our study, some comments can be made. It is suggested that optimum conditions for the use of the learning system for initial CPR instruction would be two trainees per four hour block for Heart Saver and four trainees per six hour block for BCLS. For those people initially learning CPR from the computer system, recertification is estimated to take approximately 45 minutes to one hour. Using these estimates and numbers provided by the occupational health nurse at Mobil Oil, Calgary, it would seem that to train 90 persons per year and recertify those previously taught, an instructor would spend 52 hours per year teaching CPR. This translates to $962 for this company. Add to this the cost of employee time required for initial and recertification courses and the figure increases to $9,442 per year.

If all CPR instruction was to be given by the CPR learning system, using the above time estimates, the total cost of employee time would be $3,720. The difference between this amount and the amount for traditional course employee lost hours, plus the salary of the instructor amounts to $5,722 over one year. The cost of the learning system, including duty and taxes amounts to about $30,000 Canadian. With these estimates, the CPR learning system would be cost effective for this company within 7 years. Although we are only considering the lay population at present, it would seem that such a system would become cost effective much more quickly in situations where the numbers of persons to be certified is greater.

Another supplementary finding of interest is the examination of the self-selection of people taking the Heart Saver course into either the control group or the computer experimental group. A linear probability modeling technique was used to estimate the model of selection that best fit the data. The results of this analysis indicated that the probability of selecting the computer group increased if a person had post-secondary education, was under 35 years of age and had previous computer experience. Such analysis potentially allows us to better match people to learning method.

CONCLUSION

In conclusion, the nemesis of most social scientists--no significant difference--has suggested that the criteria demanded for validity and reliability of the CPR learning system as a teaching method have been met. We can say that, in our studies conducted to date, the CPR learning system is at least as effective as traditional CPR instruction.

We will be the first to acknowledge that a study based on an N of 62 cannot be used in universal prediction. The continuation study will involve larger numbers of subjects with a variety of demographic variables, over a longer period of time.

If further study substantiates these results and the CPR learning system is shown to be cost effective, then, using the system as an adjunct to traditional instruction, we can begin to address the issues of instructor attrition, variability of instructor standards and the delivery of CPR instruction to people with limited access to traditional courses.

MATH IN BIOLOGY

An Application of Microcomputer/Videodisc Instruction

Jerry W. Edwards, Gerald M. Gillmore, William K. Harmaning,
Eric R. Halsey, and Nana Lowell, University of Washington
Gary A. Kilgore, Humboldt State University
James M. Laffey, San Francisco State University

ABSTRACT

The MATH IN BIOLOGY project has developed basic mathematics instruction which uses an interactive microcomputer/videodisc delivery system. Funded by a grant from the National Science Foundation, the instruction was principally intended as remedial mathematics for secondary and post-secondary students with career interests in the Life and Health Sciences, and especially to facilitate the access of women and minorities to educational and training programs which lead to careers in these fields. In fact, the extensive use of everyday and biological examples to illustrate mathematical principles has resulted in an attractive package for a variety of persons and purposes.

THE THEORY

Two assumptions have guided the instructional development. First, we believe that the coupling of microcomputers with videodiscs creates a powerful instructional tool. As such, the technology has particular advantages for implementing instruction which is self-paced and individualized; it requires active participation by the learner; it is illustrative and enables concrete representation of concepts and principles; and it can foster a learning environment of student control and responsibility.

Our second belief is that remedial mathematics can be effectively taught through instruction within a context area. Our materials comprise mathematics instruction within the context of biology. Motivation is enhanced through linking math skills to interests and values of the student. Learning is facilitated by concrete and relevant examples and exercises which provide visual representation of rules and principles. Generalization to performance on later tasks is improved because students practice their math skills in exercises which simulate the activities to be performed in biology laboratories or in other science coursework.

THE TECHNOLOGY

The microcomputer/videodisc hardware system with which the student interacts includes: (1) an Apple II Plus, 48K microcomputer, equipped with a language card and two disk drives, (2) a Discovision (Pioneer) model PR 7820-3 videodisc player, (3) 2 monitors (one should be color), and, (4) an Allen Communications Device (VMI board) to interface the computer and videodisc player. This hardware system costs approximately $5,000, but the interface capabilities between the Apple and an industrial player should soon be available on less costly players, and thus reduce the hardware costs by nearly $2,000.

The Apple computer was selected for the project because many schools, colleges, and training centers already own this hardware, and because of the availability of a computer-videodisc player interface device and a Pascal language system. An industrial videodisc player is used rather than a consumer version because at present the interactive functions of a consumer player are too limited for our application.

The instruction requires two monitors, one to present text and graphics from the computer, and the other for the audio-

visual output of the videodisc player. The two-monitor configuration allows the students to view computer generated display simultaneously with videodisc display. For example, the videodisc display may show rectangles being cut or colored-in by the student's keyboard activity, while the computer screen describes the problem to be solved and gives directions for cutting or coloring. Some interface devices allow for simultaneous computer and videodisc display on a single screen, but the quality of the display suffers and the amount of text which can be displayed is limited.

In general, each side of a 12 inch videodisc can hold 54,000 still images (enough space to store the Encyclopedia Britannica), present 30 minutes of motion (e.g., a half-hour television show), or contain some combination of still images and motion. The disc also has the capabilities of color video, freeze frames, forward and reverse slow motion, randomly accessed frames, and two audio channels. The MATH IN BIOLOGY videodisc has instructional material on both sides of the disc. The video includes short motion sequences produced in a studio and a biology laboratory, excerpts from films from the Encyclovideo Britannica collection, footage of the Mt. St. Helens eruption made available by two Seattle television stations, character generated frames, turtlegraphics created on the Apple, still photography, and still frames taken from videotapes of objects, flat art, and fine art. These materials were all transferred to one-inch videotape and the disc was pressed by the Optical Recording Project/3M.

Computer software was written in an instructional programming language called AVID. AVID is a set of routines written in USCD Pascal which concurrently control the Apple computer and the DVA player. The AVID language is easy to use and can be made available to instructors who wish to write their own routines or modify the programs we have produced. AVID can also be used to develop interactive programs for any "dumb" disc (a videodisc without program instructions encoded on the disc itself).

THE INSTRUCTION

The MATH IN BIOLOGY instructional package is organized into four lessons. Each lesson is further subdivided into a pretest, instruction, and a major application. The first lesson assumes only that the student can perform the four arithmetic operations on small whole numbers and can read at the junior high school level. It gives a conceptual idea of what fractions are and how they relate to the operation of division, and it provides the student with the opportunity to practice interpreting and creating fractions in interactive physical situations. In the second lesson the student learns a "key word" for multiplication, the skill of multiplying fractions, and the use of that skill in studying biology and probability.

The third lesson proceeds to the complementary skills of building and reducing fractions. It introduces percents as equivalent fractions with a "standard" denominator, and provides a great deal of practice at the estimation of percents in physical situations. Finally in the fourth lesson the student learns some practical skills for dealing with unit prices and concentrations: the understanding of ratios as concentrations is particularly important for the study of the chemical aspects of biology.

Throughout the lessons extensive examples and applications to biological topics serve to integrate and fix the student's understanding of mathematical concepts.

AN ILLUSTRATION

This section describes an instructional sequence from the MATH IN BIOLOGY curriculum. The sequence is taken from the topic on ratios and proportions.

A student sits at a microcomputer/videodisc station with two monitors. She has just completed a learning sequence about ratios and proportions and has successfully completed a test of her new skills. She is now ready to apply these skills to the task of counting red blood cells.

On the videodisc monitor she sees blood being drawn from a patient's arm and learns something about why blood may be analyzed. She is then told about counting red blood cells and introduced to a hemacytometer. A hemacytometer is an apparatus similar to a microscope slide containing a grid, which is used to determine the number of red blood cells in a defined unit of volume. After an animated microscopic flight under the coverslip to see the grid and an illustration of a unit of blood with blood cells, she returns to meet three videostudents who are preparing to count red blood cells.

The videostudents are in a biology laboratory and have a microscope,

hemacytometer, red blood in a test tube, diluting fluid and extra test tubes. One of the videostudents puts some blood on the hemacytometer and the videostudents look through the microscope. By pressing "M" on the keyboard our student also sees the view through the microscope. The students find that they cannot count red blood cells in this red smear, and one of them suggests diluting the blood.

Our student and the videostudents progress through several dilutions, at each point observing the slide and finding that they are unable to count the cells, but recognizing progress toward their objective. Our student participates in the work by controlling the amount of diluting fluid released from a pipet and observing the effect of various levels of dilution. After several dilutions, the students reach a concentration which allows them to count the red blood cells within the grid of the hemacytometer. Our student works through a series of calculations, first averaging the number of red blood cells in several sections from the hemacytometer grid and then calculating the number of red blood cells in a unit of the dilution. Lastly our student uses knowledge about the ratio of blood to diluting fluid and establishes the red blood cell count for the original blood.

In this exercise the student brings to play the math skills learned in an earlier tutorial and sees the relevance of these skills to her career interests. In addition she gains experience at working in a simulated biological laboratory.

CONCLUSION

The MATH IN BIOLOGY project has produced courseware which utilizes advanced technology in an effort to solve an important and unremitting educational problem. The technology is available today and steadily becoming more affordable. The challenge to educators is to use this technology in wise and creative ways to enhance student learning. While complete evaluation of MATH IN BIOLOGY remains to be done, we believe that this instructional package is a creative approach to developmental education and a step in the advancement of educational technology.

Interactive Video - The Next Frontier

Dorothy J. Fishman, R.N., Ed.D. - Interactive Learning Systems

ABSTRACT

Interactive video, the integration of computer and video technologies, has added multi-sensory, multi-level dimensions to education.

This paper discusses the viability of developing individualized computer-assisted video mastery learning packages from scratch or from existing instructional videotapes.

The development and field testing of an interactive video module teaching cancer chemotherapy to nurses, developed at the Massachusetts General Hospital in Boston, Massachusetts, will also be discussed and demonstrated.

Research results indicate that interactive video increases achievement and retention when compared with instructional video and lecture. Interactive video appears to be the next frontier in education and a challenge for the curriculum designer.

Interactive Video - The Next Frontier

Imagine a technological advance in information processing that adds a new dimension to how people learn. An individiualized multimedial system that has a uniquely exciting potential for nursing education. A system that allows the learner to see and hear reality while mastering complex concepts and skills. A dynamic learning system that provides immediate feedback and guides remediation while the learner controls the learning process.

If this sounds like something out of "Star Wars" . . . think again. The technology is here today and it is called interactive video or computer-assisted video instruction.

Interactive video is the marriage of video and computer technologies. It integrates the computer's logic with the color, sound and motion of video. It creates an interactive, individualized learning system that is user-paced, user-sequenced and hopefully user-friendly (Fishman, 1984).

Whether videotape or videodisc, "interactive video creates a synergism between man and machine that converts the passive learner into an active participant in the learning process" (Fishman, 1984, p. 17).

Instructional television is designed in a linear format. The program is viewed from beginning to end in real time without any interruptions. The relatively passive viewer has no control over the sequence, pace or selection of content.

Interactive video, on the other hand, is designed so that the user must actively interact with and control the learning process.

Feedback can be developed on several learning levels. This allows the slower learner more time for explanatory information, while the faster learner has the ability to advance at will.

Interactive video, therefore, is designed for multi-level, multi-sensory learning to enhance motor, affective and cognitive skills.

The potential of this new medium to foster mastery learning is exciting. However, we must be realistic and ask the question: Does it Work?

To answer the question, a research study was designed to include the development and field testing of an interactive video program using an existing instructional videotape. During the field test interactive video was compared with the traditional classroom lecture and linear video methods of cancer chemotherapy instruction.

A review of the current literature shows few research based studies showing the effectiveness of interactive video. However, the literature does identify the following conditions of learning: feedback, cueing, practice trials, remediation, individualized learner control of pace and sequence, and active learner participation.

The interactive module was developed with the cooperation of the Television and Nursing departments of the Massachusetts General Hospital (MGH) in three phases.

Phase 1 - the selection of the videotape "Cancer Chemotherapy," produced by MGH. The videotape is the winner of the 1981 Biological Photographic Association's Medical Education Award.

Phase 2 - infrastructural analysis. This included the evaluation of the technical and production quality of the program, and it sought to answer the question: What is the structure of the instructional process? A content analysis of the entire text analyzed and specifically identified the content. The syntactic and semantic features were then evaluated at the sentence and whole text level. Based on the content analysis the videotape was segmented into four sections which contained eight chapters.

Section	Chapter
I. Advanced Organizer (Objectives)	1. Introduction (History)
II. Information (Content)	2. Treatment Approaches
	3. Cell Cycle
	4. Uses and Limitations
	5. Drug Classification
	6. Drug Selection
III. Patient Interviews	7. Side Effects
IV. Post Organizer	8. Summary

Since visual education should be based on the premise that every picture is a statement (Arheim, 1960), the fidelity of the visual sequences used to represent the educational concepts being transmitted to the learner were evaluated. The results of the evaluation showed that 95% of the visuals were related or marginally related to the content assertions (Fishman, 1984, p. 18).

Phase 3 - Development of the interactive module. This included the instructional design for both the computer and video segments, the multiple choice questions, remedial feedback loops and the interactive computer program.

By designing and using a comprehensive storyboard which included text and visuals we were able to map the one-to-one relationship between the objectives, content and test questions, in addition to visualizing the relationship between the video and computer segments. The last step in the process of design was the development of the computer program.

A smooth transition between computer and videotape player was insured by allowing the computer to control the synchronization between the computer-generated text and the video segments of the program. By electronically recording sequential frame numbers on the videotape, the start and end points of any video frame or segment was easily identified and accessed. This also allowed the computer to be used for editing videotape segments that were not included in the finished interactive program.

Now let's walk through the interactive video program. By inserting a floppy disk into a disk drive, and the corresponding videotape cassette into a VHS videotape player, the learner begins the program. Instructions are given via computer generated text. The user views a video segment, is able to repeat the video segment as many times as needed before being tested, is then asked a multiple choice question. If the question is answered correctly, the next question or next video segment appears. If the question is answered incorrectly, that portion of the videotape containing the correct answer is played. The learner is then retested for content mastery before being allowed to proceed through the program.

"Knowledge is acquired, verified and permanently recorded" (Fishman, 1984, p. 22). The answer summary is stored on a disk and a paper printout is readily available.

The interactive program was field tested using an experimental pretest-posttest-retention design.

The study was concerned with the differences in student achievement, retention of cognitive knowledge, learning time and nurses' opinions.

Sixty-four registered nurse volunteers from a large teaching hospital were randomly assigned to one of three instructional groups: linear video, traditional lecture or interactive video.

> The nurses' ages ranged from 22 to 56 years with a mean of 32 years; work experience ranged from 0 to 32 years with a mean of 10 years. Education levels included: seven diploma graduates; six associate degrees, 37 baccalaureate; 10 masters degrees and four doctorates. Seventy-two percent of the nurses had no prior computer experience. (Fishman, 1983)

All treatment groups received the same content material and followed the same sequence of events. Day 1, each group answered a demographic questionnaire, took a pretest, received a designated instructional treatment and answered a posttest and an Opinionnaire. Two weeks later each group answered a retention test.

The traditional classroom lecture group received the same content as the other two groups. The instructor was the same on/off camera talent used in the videotape.

The interactive group received the cancer chemotherapy computer-assisted video instruction. Each nurse interacted with the program via a computer keyboard.

The linear video group viewed the video tape "Cancer Chemotherapy" from beginning to end without interruption.

Three measurement instruments were used to obtain data. The cancer chemo-

therapy cognitive test consisted of 30 multiple choice questions. The same questions were used for the pretest, posttest and retention test. Only the order of the questions and answers were changed. The internal consistency reliability for the test was estimated at .83, using Cronbach's Coefficient Alpha. The Opinionnaire was used to assess nurses' opinions toward the instruction and the conditions of learning. This included a semantic differential survey (Coefficient Alpha, .72) and a 7-point Likert Type Scale, Alpha .77 (Fishman, 1983).

An analysis of covariance was used to determine the significance of the differences among the mean scores of the three treatment groups on the cancer chemotherapy cognitive test.

The dependent variable was the nurses' posttest scores on the test; the covariate was the pretest score on the same test; and the independent variable was the instructional group.

The adjusted protest-posttest gains showed that the interactive group (92%) did significantly better than either the linear video (78%) or traditional lecture groups (74%). Post Hoc comparisons showed no significant difference between the lecture and linear video groups, which supports previous research findings (Fishman, 1983, 1984).

The adjusted pretest-retention gains revealed that the interactive video group (83%) did significantly better than the linear video (74%) or lecture groups (70%).

The retention scores of the interactive group were higher than the posttest scores of the other two groups.

No statistical tests were performed on the learning times because there were no within-group variations for the linear video (27 minutes) or the lecture (50 minutes) methods of instruction. The interactive learning time mean was 54 minutes. Although linear learning time was almost half of the interactive learning time it is deceptive because of the vast difference in posttest and retention test results. It really suggests that the student had control of the interactive instruction and no control of either the lecture or linear video method of instruction.

We asked the nurses to rate the seven conditions of learning on the Opinionnaire.

Conditions of Learning

1. Content presented too rapidly
2. Learned at own pace
3. Items related to content
4. Content not appropriate
5. Trouble focusing attention on material being presented
6. Would recommend instruction
7. Instruction allowing for feedback

Only four of these items were significant.

Item 2 - people learning at their own pace is self-explanatory.

Item 3 - items related to content. This is reasonable because both interactive and linear groups had the same content.

Item 5 - the interactive group had more trouble focusing their attention on the material being presented because they had control over the pace of exposure. The other two methods are outside of student control.

Item 7 - reinforces what we already know about the benefits of positive feedback.

We also wanted to know if there was any relationship between achievement and age, education or work experience. The only significant finding was that the nurses who held masters and doctorates had higher achievement gains and the older, more experienced nurse found the computer keyboard more intimidating (Fishman, 1983).

We asked the nurses in the interactive group to list the major strengths of interactive video. The conditions of learning they identified were immediate feedback, remediation, the ability to control their own learning, and work independently.

When we asked the nurses to list the weaknesses of interactive video, they identified the inability to ask questions of the instructor as the major weakness. However, this was reported by only 27% of the nurses (Fishman, 1983).

There were several observations made by this investigator during the

course of the study. I learned the hard way that interactive video curriculum, storyboard and script are designed very differently from linear video and has little in common with a traditional lecture format. Some interesting things that I learned were:

1. The use of dissolves and split edits though very common in video and slidetape productions can cause major problems when they fall at points of interaction.

2. Extra video footage is helpful for additional remedial loops and should be planned for before shooting begins.

3. Storyboarding the entire interactive program is a must. It avoids costly errors in time and money.

4. Knowledge of the exact retrieval time of the remedial video segment is very important when using videotape. It allows you to design your computer program so that access time is kept at a minimum.

5. Screen design should be kept simple. Bouncing balls and cluttered screens are learning distractors.

And, the last step in the development process should be the computer program. This will not only save time but it will also save money.

There are many advantages to using interactive video learning systems. "We can no longer assume that verbal descriptions and the blackboard are the prerequisites to learning" (Fishman, 1984, p. 22). The findings of this study indicate that interactive video increases achievement, and retention over linear video and lecture methods of instruction provides individualized instruction that is learner controlled, enhances mastery learning, and nurses' opinions toward this technology are positive.

The potential viability of developing computer-assisted video programs from scratch or from existing videotapes offers an efficient opportunity to create individualized learning programs that can be used in clinical and academic settings, at the convenience of the learner, with or without the presence of an instructor.

Interaction video appears to be the next frontier in education and a challenge for the creative curriculum designer.

References

Arheim, R. (1960). Visual thinking. Los Angeles: University of California Press.

Fishman, D. (1983). Development and field testing of a computer-assisted video module teaching cancer chemotherapy to nurses. Unpublished doctoral dissertation.

Fishman, D. (1984). Development and evaluation of a computer-assisted video module for teaching cancer chemotherapy to nurses. Computers in Nursing. pp. [illegible]-23.

FROM NOVICE TO "EXPERT"

IN THE DEVELOPMENT OF A VIDEO DISC

Gloria Lombardi, County College of Morris

ABSTRACT

This paper describes the experiences of team members at the County College of Morris in their first attempt to design and develop a Level III CAV* videodisc to be played in conjunction with a Sony SMC-70 microcomputer. The paper discusses the hardware/software configuration and the importance of the videodisc to enhance student nurses' ability to exercise and implement their clinical judgement freely. The project is discussed from the program design, through the video production and the programming stages.

*A Level III program integrates a videodisc with a microcomputer. CAV (Constant Angular Velocity) allows the option of freezing or parking on a frame for an indefinite period of time.

Introduction

Teaching the process of clinical problem-solving is one of the greatest challenges that nurse educators face. Its difficulty is perhaps surpassed only by the problem of evaluating how well a learner understands the complexity of the clinical situation. Considerations of patient safety, in traditional hospital settings, usually do not allow learners the option to exercise and implement their clinical judgements freely or to experience the full consequences of their decisions.

In an attempt to aid students with their clinical decision making, this author initially developed sequential patient management simulations to which the students responded in writing by asking for further data. Feedback was then given to students in response to their requests through a set of information cards from a bank of previously developed data. Based on data collected, the students made decisions about the situation. Although this method was interactive and allowed students to reach conclusions by a variety of widely branching routes, it was costly, time consuming, and difficult to administer to more than a few students at a time.

It soon became evident that this manual system could be made much more efficient and interesting for the student if computerized. In addition, the author believed that if students had the opportunity to actually view the situation, the simulation would much more closely approximate reality. A search for the technology that would incorporate these variables led the author to decide that her simulations could be most effectively developed by designing an interactive videodisc.

A project was planned and a grant proposal identifying the need, the target audience, objectives, and a plan for evaluation was submitted and jointly funded by a Computer in Curricula grant from the New Jersey Department of Higher Education and by County College of Morris. Since the project that was developed merged the tools of both television and the computer a team approach was required. The team consisted of content experts in nursing who were responsible for the design of the project, television production experts responsible for videotaping, and a computer expert responsible for programming. Because we lacked expertise in programming a videodisc using the

Sony Computer, it was decided to contract out programming responsibilities.

Selection of Delivery System

The first major task the team faced was the selection of a delivery system. Because we were producing a Level III CAV videodisc, we needed not only the laser disc player and color monitor, but a computer. There were many configurations available to choose from. The majority of the grant money was allocated for equipment, and the only consideration was--which equipment would be best for the needs of the project. We chose a complete Sony system for several reasons. This was our first videodisc production and we needed equipment and software available that allowed for the creation and filming of computer-generated text and graphics. The ability to do this was important to our project because of the decision to include a client's chart on the videodisc. Another important feature was the ability to superimpose computer-generated graphics and text over the videodisc images on the same monitor. We are pleased with our delivery system, however, today there are many more options available in this rapidly expanding field.

Design

The purpose of the project was to create a simulation which would allow the learner to exercise options, implement decisions, and follow-up the results of those decisions free from the constraints imposed in most clinical teaching situations. Through recording of responses by the computer, an instructor would be able to review and evaluate the learner's performance in the simulation.

The most important aspect of the content designer's role is to communicate ideas to the video producer and the programmer. A flow chart served as our communication tool as well as a check on the logical flow of the program because of the branching. There are no standard symbols for videodisc motion and video disc stills. Therefore, the team designated symbols.

Figure 1 illustrates a portion of the flowchart as well as the designated symbols.

In a linear production, the viewer watches the drama unfold with a beginning, middle, and end. The learner in an interactive production becomes an integral part of the momentum or movement of the drama. Therefore, the video segments are a variety of short segments. It was critical that all alternatives chosen branched to a response. It was necessary to devise a scripting format that would include video segments using the two audio tracks, as well as computer screen design for the menu choice and graphics.

Figure 2 illustrates an example of the scripting page that corresponds with one of the flowchart symbols.

The instructional competency states that the learner will utilize interactive videodisc simulation to become more proficient in clinical decision-making by meeting the following objectives:

1. The learner will receive visual and auditory clues from a patient care situation.
2. The learner will utilize previously acquired knowledge to interpret and analyze clues.
3. The learner will choose alternative actions based upon his or her perception of the situation.
4. The learner will be shown the consequence of his or her decisions.
5. The learner's performance will be recorded and evaluated.

The videodisc depicts four postoperative situations and a client chart. Mrs. Morris, a postoperative cholecystectomy patient, presents symptoms common to many patients following abdominal surgery. Using the disc, the learner collects clinical data, analyzes the situation, makes decisions, and sees the results of those decisions depicted on the screen. Following the branching paths of the options presented, the learner actively applies and builds on his or her clinical knowledge to achieve a satisfactory outcome for Mrs. Morris.

The system is designed to utilize two floppydisks. One diskette controls the program. Each student is assigned a diskette that records his or her answers. Each clinical instructor is able to get a printout of a student's responses. We are also able to gather information for program improvement and further development of decision making skills. Eventually, the system will include a quantitative method of evaluating the student's ability to problem solve so that a student can get instant feedback as to his or her progress.

INTRO
(SEGMENT 1)

FIG. 1

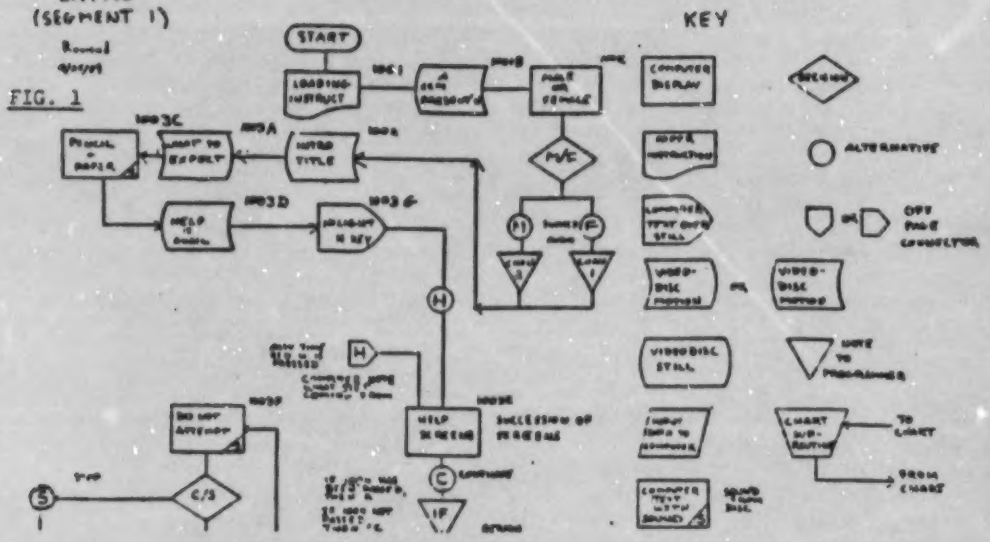


FIG. 2

REVISED 9/25/84

1003D

Segment # 1

Video Disc [1] Computer [1]	Segment Title: Intro
Motion X Still ____ Time ____	Help is avail.
Frame 02245 to 02620	Visual Video Disc
Text over still ____	Picture of Sony computer keyboard
Branching From 1003 C	
Branching Go To 1003F	
Graphics	Visual Computer
Translucent window highlighting "E"	

Audio 1 2
(1) Male voice
(2) Female voice

THERE IS A "HELP" MODE AVAILABLE FOR THIS PROGRAM. ANY TIME YOU WISH HELP WITH THE DIRECTIONS LOCATE AND PRESS THE RED H. DO IT NOW.

BEST COPY AVAILABLE

Video Production

Videodiscs are capable of high quality sound and pictures. The industry recommends one-inch-high band-color VIR production. Because we wished to make an inhouse production, we had to use 3/4U. Therefore we had to worry about generation losses. (If raw footage is recorded then edited together it becomes second generation. Then if a work copy is made it becomes third generation. This adds a lot of undesirable qualities to the final picutre.) We kept the losses to just two generations. We edited directly on the master tape. We also had to watch the video level throughout the production because videodisc mastering cannot tolerate high video levels.

Because the frames of the videodisc have to be addressed later by the computer, a SMPTE time code had to be laid down on the master tape. We used a non-drop-frame SMPTE code that keeps an accurate frame count. This code gives a frame number to each video frame on the disc. There are 30 frames for each second of motion on the disc.

We wanted the learner to "see" himself or herself in the situation. Therefore we used the subjective viewpoint. The segments were taped as if the viewer were a participant in the action. The viewer is constantly prompted and consulted visually and aurally. Taping in the subjective viewpoint provided interesting challenges for the production.

We needed to put chart information on the videodisc. This had to be done in the form of text generated from the computer. Presently, digital information (computer output) cannot readily be put directly on videodisc. It must be converted into video signal first. The American NTSC television standard had to be used. Fonts had to be high quality for legibility when on tape and to withstand the resolution lost with each generation of editing. Sony's video titler software package was used to generate the computer text. We had to tape samples of colors because certain colors bled and made text unreadable when converted to tape.

Viewers are accustomed to high quality broadcast video productions. Therefore, to enhance our production, we decided to use some special effects. For example, the movement of the video from scene to scene looks, to the viewer, like the turning of pages. All special effects were done off campus at a post production facility.

Programming

We had previewed authoring systems for interactive video. At the time of the development of our design, we did not find a system that allowed us the flexibility needed to handle the complexity of our design. BASIC was used for programming, making it easier for the rest of the team to later edit or modify the program.

Other than dealing with the communication links between the systems, most of our program instructions involved asking for an input and acting on the value received from it. The greatest challenge was to design a diskette that recorded student responses.

Conclusion

The project took approximately one year to complete; however, all of the team members worked on the project while maintaining their regular college responsibilities. The program will be tested by our students during this fall semester. This author will be field testing the program in the spring semester.

References

Bejar IS. Videodiscs in Education. Byte 1982; 6: 78-104.

Deblois ML. Videodisc/Microcomputer Courseware Design. Educational Technical Publishing, NJ, 1984.

Fort WM. A Primer on Interactive Video. AV Video 1984; 10:39-41

Yampolsky ML. The Next Generation of Interactive Videodisc. E.I.T.V. 1983; 6: 40-44.

BEST COPY AVAILABLE

LESSON DESIGN, AUTHORING, and VIDEO ISSUES IN THE PRODUCTION OF INTERACTIVE VIDEOTAPES

John F. Moore

Virginia Polytechnic Institute and State University

ABSTRACT

There are many design and production steps which must be managed in order to create effective interactive videotape (IV) courseware. This paper describes what was learned by the development team during a project to produce two IV independent study lessons. Emphasis is placed on project planning, various lesson design considerations, unique video production issues related to tape-based IV, and the advantages and limitations of using an authoring system. Delivery system enhancements are described.

Introduction

A lesson development team faces a number of design and production steps which must be managed in order to create effective interactive video (IV) course materials. This paper will describe what was learned from creating two interactive videotape lessons for use by students in an independent-study format. Project planning, the use of an authoring system, lesson design, and video production considerations will be discussed.

The purpose of the project described was to explore the development and delivery of instruction using interactive videotape. The project development team included the client (the course instructor), an instructional developer, and a video director. The hardware used was a Sony Betamax SLO-325 videocassette recorder (VCR), an Apple II+ computer with color monitor, and a BCD Associates model 450 interface card.

Project Planning

During the planning phase, there was considerable discussion about both students' learning needs and the client's course management requirements, with an eye towards how an IV system could be of use. The informal needs assessment served as a focal point for many subsequent decisions, although compromises became necessary in order to cope with hardware and software limitations.

As the lesson objectives evolved, gradually we began to synthesize the development methods used in some of the pioneering videodisc projects (see, for example, DeBloois, 1982). An early decision was to determine an efficient method for the production of the computer programs each lesson would use.

The Authoring Alternatives

The computer programming needed for an interactive video project can be done using either a language (such as PASCAL, SuperPILOT, or BASIC) or with an authoring system. Steinberg (1984) describes the tradeoffs involved with each. The choice of the production method for generating the computer programs can affect the speed and ease of lesson creation as well as later maintenance.

A chief advantage of using a language is that the instructional strategy, program branching, answer

processing, and screen layouts can be customized to the extent determined in the lesson design phase. Compared to an authoring system, fewer compromises become necessary. While this versatility may be desirable in many cases, the programming effort required to implement the lesson can consume a considerable amount of time and money. Furthermore, as evaluation information is gathered, making changes and improvements in the lesson can become a laborious task. Even though the benefits of using a language can be considerable, projects often have neither the time, staff support, or funds needed to utilize this method.

An alternative method of production is to use an authoring system. The user does not need to know a computer language since the authoring software asks questions and takes responses in English. Once the lesson flow and screen layout decisions are planned, much time can be saved by using this method. However, the ease of program generation by using an authoring system is often counterbalanced by a less flexible lesson design. Examples of such limits on the lesson designer's options can occur in the layout of information on text screens, menu branching, types of questioning strategies, and the degree of student control of lesson flow possible. As more advanced authoring systems reach the market, these limitations are gradually disappearing.

A key factor in our decision to use the BCD Associates authoring system was the efficiency it would add to the production phase of the project. In order to produce the lessons on a reasonable schedule and control costs, we accepted the limitations on lesson design. There was hope that some modifications could be made which would partially alleviate the authoring system limits.

Features of the Authoring System Used

BCD's "The Instructor" (version 4.2) is a moderately powerful authoring system. Its major functions include: 1) an authoring program (AQ 3.1) to generate computer "pages" and their branches; 2) an operating program (CAVOS 4.2) which selects and retrieves the "pages" from disk, controls the VCR, and maintains student records; and 3) a tape logging program (Logger II) to tag video scenes for retrieval. The page formats are: information, multiple-choice question, keyword match question, graphics, menu, random page select, audio-only, video/audio, and VCR record. There can be up to 100 "pages" and 100 video scenes in one lesson. Any page can branch to any other page and/or video scene(s).

Among the system's limitations was no convenient way to show video and ask a question simultaneously, as would be useful for visual discrimination tasks. Conditional branching is not implemented easily, nor is there an easy way to clone a page. Since each page actually consists of several character strings, lessons are paused frequently while the Apple memory string storage area is cleared. There are also minor limitations in features like screen formats, lack of graphics animation, the preset number of answer choices used for multiple choice questions, and the typically slow retrieval of text pages. Nevertheless, "The Instructor" is a moderately powerful, although somewhat dated, tool for creating and maintaining lessons.

Lesson Design

A large investment in planning time at the beginning of each project helped minimize re-design and re-editing later. Developing the IV lesson scripts was a lengthy and meticulous task. After establishing the overall structure for each lesson, the learning goals and tests for each unit were developed. Planning cards, used in a storyboard-like fashion, showed one computer frame or described one video scene. Detailed information about page numbers, scene lengths, and branching data for each exit node was also recorded on the cards. These cards could easily be rearranged and altered as the lesson units were "fine tuned."

As each unit was being developed, a flow chart also was drawn to show lesson structure and "page" components. First the mainline path and then each branch was planned, scripted, and drawn. The flow chart was invaluable as it let the developers easily visualize the overall lesson structure and inspect the various paths a student might take through the material. During the debugging phase, the charts were also used to trace programming and frame composition errors.

The project development team realized that because of the lack of

custom programming flexibility, minor adjustments in the instructional design preferred would likely be unavoidable. Some of these concerns and changes are described below.

One of the early goals was to provide students with considerable flexibility and a sense of control while studying the lessons. By providing a menu at the end of each lesson unit, several path choices were made available. These choices might include: 1) go on to the next unit; 2) review a part of this unit; or 3) go back to the main menu.

Another consideration was the built-in limit of 100 pages and 100 video scenes. Such a capacity may seem sufficient until different questioning alternatives are considered. Assuring mastery of some content, for example, may require a relatively complex design and thus many pages. Nevertheless, a designer is urged to first plan as if there are no such limits and then cut back or redesign only if necessary.

The "best" test and remediation strategies are often situationally determined by individual lesson objectives. For example, a simple strategy is to branch all the wrong answer choices for a single question to the same remedial page, and then go back to the original question for another try. Programming like this uses few pages but often may be considered instructionally weak. Having only a common remedial page may make it difficult for much clarifying information to be given to a student, especially if widely different misconceptions are used as wrong answer choices.

An alternative design strategy is to provide a separate remedial page especially written for each wrong answer choice. A different (secondary) question on the same topic may then be posed. This kind of testing and remediation may be quite useful in certain lessons. While intuitively more powerful, in other lessons it may be unnecessarily complex. Remembering the learning and testing objectives of the lesson can help lesson designers carefully manage their "page" resources.

A final adaptation example concerns asking questions which have a visual component. Showing static figures or pictures in a workbook is a common solution to this problem. However, for questions which require moving or dynamic images to enable visual identifications or comparisons, a static picture in a workbook is clearly not sufficient. Because the authoring system did not support simultaneous video with answer acceptance, an alternative method was developed. For example, a short video comparison of two procedural methods is followed by a menu page. The menu page, functioning like a question page, displays the question "Which method produced the second result?" Included on a list of answer choices is an option to see the comparison again. While this technique is clearly a compromise and relies on short-term memory, it does permit limited use of dynamic visual testing.

The development of this method of testing illustrates the kind of unique instructional opportunities which can be built into an IV lesson, regardless of authoring system limits. During the design meetings, it was generally more productive to disregard the acknowledged system limits and strive to plan the "ideal" instructional unit. Then, after several days, the script would be reexamined and adapted to the authoring system. This work pattern served to encourage more creative lesson and presentation designs by separating the conceptual planning from the implementation.

Video Production Issues

Iuppa (1984) gives details about the need for careful scene arrangement in the preparation of videodisc master tapes. Due to the longer search times in a tape-based system and the possibility of tape edge damage during thread-up, scene design and layout is important for continued efficient operation.

Remedial scenes often are actually edited segments from previously seen (mainline) scenes. Ideally these should be edited as separate segments (stand-alone remedials) rather than accessed as "embedded" sub-scenes within mainline scenes. Doing so can minimize damage caused by repeatedly threading and unthreading the tape at the same points (the start and end of an "embedded" remedial). Second, search time is minimized because the tape does not need to be rewound to play an embedded remedial. Third, a stand-alone remedial scene can incorporate a different treatment, even though it uses the same basic video as the mainline scene. Different captions or problem-specific narrations can thus be used.

Other unique problems can occur during editing. One is the identification, re-narration, and editing of stand-alone remedial segments. Consequently, much original footage often has to be edited in several slightly different ways. This can increase editing time substantially. To help the video staff understand the apparent lack of program continuity during editing, provide them with a full script, flowchart, and final tape sequence chart.

Delivery System Modifications

After the tape for the first lesson was produced and debugging was underway, it became desirable to slightly modify the BCD CAVOS operating program. (The substantive changes which had been originally desired in the authoring program were not made due to the complex nature of that program and inadequate documentation.) These changes helped increase the apparent power and ease of use of the system.

One enhancement was to give the student more control over the sequence and flow of the video playbacks. The added option will stop and back-up a scene 20 seconds, then restart it. This is a convenient way to review information, without needing to watch the entire scene again.

Several other enhancements were made to improve communications. Line 23 on the Apple text screen was reserved by BCD for system messages such as "Press return to continue." All such messages were switched to inverse print in order to isolate them more from the lesson text. Another alteration was to give the message "Searching (about n seconds)" when a tape search will exceed ten seconds. Early expectations that the time for tape searches could be "masked" by involving the student with a screen of text or a workbook activity proved unrealistic over an entire lesson. A long search is often unavoidable when a student wishes to review a scene some distance away on the tape. Knowing how long a search will take may encourage a feeling of control over the system and alleviate a student's feeling of not knowing quite what the system is doing.

Summary

Producing interactive videotapes involves many steps, including determining lesson design; planning scenes, pages, and branch paths; video production; and generating and debugging the computer program(s) needed to present and manage the student presentation. Adapting the lesson to the structure and limits of an authoring system may require compromises in the instructional design. Those who are beginning to create IV lessons should carefully consider ways to expand their designs beyond the perceived limits of their system. Plan carefully and think non-linearly. A willingness to experiment with both lesson design and implementation can result in an improved product.

References

DeBloois, M. L. (1982). Videodisc / microcomputer courseware design. Englewood Cliffs, NJ: Educational Technology Publications.

Iuppa, N. V. (1984). A practical guide to interactive video design. White Plains, NY: Knowledge Industry Publications.

Steinberg, E. R. (1984). Teaching computers to teach. Hillsdale, NY: Lawrence Erlbaum Associates.

DESIGN OF A SOFTWARE INTERFACE

FOR AN INTELLIGENT, INTERACTIVE, AND INTEGRATED VIDEODISC SYSTEM

Darleen Pigford - Indiana University at South Bend

Albert E. Miller and Eugene W. Henry - University of Notre Dame

ABSTRACT

The purpose of this paper is to convey a design for a software interface for an intelligent, interactive, and integrated videodisc system. The interface allows interaction between videodisc player and a stand-alone personal computer and provides a basis for the superposition of additional interfaces with other peripherals. The design is portrayed as a three level hierarchical chart and each functional module is discussed in terms of structured flowcharts and structure charts.

INTRODUCTION

The theme of this paper is a description of a software design for an intelligent, interactive, and integrated videodisc system. The first part of the paper is an overview of videodisc. Details, definitions, and background information are given on videodisc, its nature and operation.

The second part of this work details a software design for implementing the desired videodisc system. The problems and constraints are discussed before the detailed explanation of a three level hierarchical design for the videodisc interface. The design is explained with structured flowcharts, structure charts, and examples of usage in pseudocode. Using the interface to allow asychronous input/output is included. The concluding remarks identify the implementation of the design and emphasize the purpose of the design: flexibility for utilizing videodisc system for business or training purposes.

VIDEODISC: AN OVERVIEW

Few technologies emerging on the scene in the past decade have caused as much interest and speculation as that of interactive videodisc. This new revolutionary medium is one of the fastest growing areas of communications for the 1980's New opportunities arise because of the combined promise of computer control and video presentation.

In general, a videodisc is an extremely dense storage ddvice capable of storing combinations of moving and still visual images, high quality audio, and digital information. Each side of an optical laser videodisc - which has the appearance of a shiny record - can contain up to sixty minutes of real-time linear programming on an extended play disc (constant linear velocity or CLV). If the slower standard speed (constant angular velocity or CAV) is used, the maximum is thirty minutes per side. The power of these 54,000 frames on the videodisc is that each frame is individually accessible within a matter of seconds. Once these images are on the disc, however, they are presently not changeable or alterable.

The principles behind the operation of a laser videodisc are simple. Analogue signals are encoded in microscopic pits under a transparent coating and read by a laser pickup that senses the difference in reflectivity between the pits and the background. This pickup can be positioned anywhere over the surface of the disc very quickly because it never touches the disc. It is similar to a floppy disc, except that it is read-only and it holds much more data.

In this paper, moreover, intelligent videodisc will mean the interfacing of a

videodisc player to a computer. This harnessing of the videodisc with a computer opens many new possibilities for training and business usages. In order to use a videodisc, a participant or trainee must have access to a videodisc system. This configuration (training or work station) has five components: 1) a videodisc player, 2) a computer to control the videodisc player, 3) input device(s), 4) output device(s), and 5) the software to drive the system.

The emphasis of this paper, moreover, is on "interactive videodisc". This term means that a user of the videodisc system can control the playback by chosing portions at random. This is contrasted with linear viewing which starts at the beginning and proceeds in a straightforward and unchangeable manner to the end. Movies and TV programs are examples of linear systems, the antithesis of interactivity. This interactive system, moreover, will be integrated: that is, it will utilize a variety of input/output devices like bar code, touch sensitive screens, random access audio, etc.

A logical unit of instruction, training or presentation can be visualized as a state diagram (Figure 1) of four fundamental states: start(S), information(I), response (R),and exit(E). Reading the diagram from left to right, one can model a logical unit of instruction as any finite combination of I states and R states with one initial start state (S) and one exit state (E). For example, a logical unit of 35 videodisc images followed by a question which requires a participant's response (or interaction) could be modeled as S, followed by I, followed by R, followed by E.

The information state can be conceptualized as a page of text, text over a graphics screen, videodisc images with sound, or any other combination of media. The information state could be as simple as a sentence from a page of text or as complex as an entire chapter in a book. The response state denotes that an action is required from the viewer or participant. The response, usually from a question, could lead to more information, to another question, or to an exit of the logical unit. (The controls generated from the response are not shown in the state diagram). Both the information and response state could be composed of any media that the author desires.

Hence, the videodisc system is intelligent (interfaced to an external computer), interactive (controlled by a participant), and integrated (composed of combinations of media).

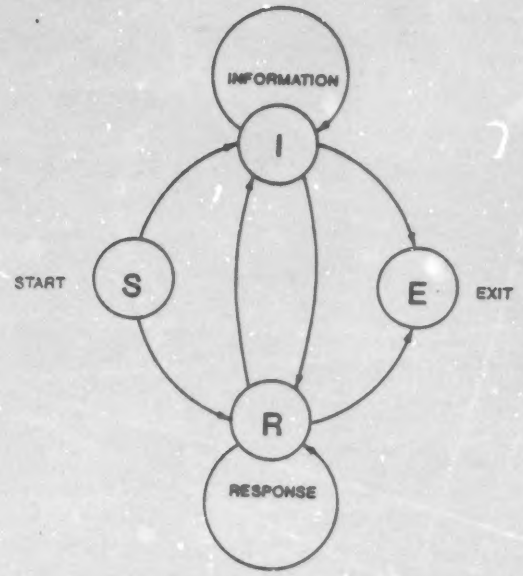


Figure 1. State Diagram of a Logical Unit for a Videodisc System.

SOFTWARE PROBLEMS

In order to produce videodisc packages for an intelligent, interactive and integrated videodisc system, an author (creator) would need a set of programs to produce the desired results. Ideally,the software needs built-in interfaces to other peripherals (light pen, joystick, bar code, random access audio, touch sensitive screen, tablet, etc.). The designer of the videodisc modules should be free to select and/or combine any of the media possible from these peripherals. Unfortunately, commercial authoring systems and authoring languages do have this total flexibility. For example, one could find a commercial authoring system to incorporate videodisc with touch sensitive screen, but not with bar code. In order to provide the maximum flexibility in videodisc systems, the Interactive Videodisc Research Laboratory at the University of Notre Dame designed and implemented a set of subroutines or procedures. This interface is flexible, usable, and enhanceable with additional interfaces for other peripherals. The entire thrust of this paper is to convey the design as a fundamental or basic augmentable interface. (Any reference to actual code will be made in pseudo code.)

DESIGN CONSTRAINTS

There are some constraints for this design. First of all, the videodisc

player is intelligent. This means that the videodisc has its own microprocessor and is capable of handshaking with a stand alone personal computer. In other words, both the personal computer and the videodisc player can send and receive messages. The physical interface between the two pieces of equipment is a cable between a port on each apparatus.

Another assumption concerns the operation of the videodisc player. The player will acknowledge each command from the personal computer and any time-dependent command (search or play until) will be further acknowledged by some additional signal to identify completion in real-time. The player, moreover, can work in stand along mode via its own control panel. Internal operations of the discplayer can be displayed on a window overlayed on the monitor.

DESIRED PERFORMANCE OF THE VIDEODISC INTERFACE.

The videodisc interface must allow a programmer to access the videodisc player without any knowledge of the technical operation of the machine. The designer/programmer has two basic tasks to perform once the instructional design and videodisc are ready. A correspondence must be made between still frames and their unique frame numbers and between motion sequences and their unique start frame numbers and stop frame numbers. A still frame is one videodisc image that is displayed continuously at 30 frames per second (on a CAV disc) and a motion sequence is the sequential playing of frames at 30 frames per second from a given start frame to a selected stop frame. These frame addresses, moreover, usually range from one to 54,000 on each side of a videodisc. The second task for the programmer, moreover, is to code a program to make the videodisc display the images as desired for the videodisc system.

Special software to accomplish the first task is not essential. The discplayer can be operated with its own keypad in stand alone mode (external CPU off). To identify frame numbers with desired images, a designer/programmer can manually operate the disc player and reveal the frame numbers being displayed. This is accomplished by pressing the appropriate key on the control pad. This key superimposes a small window on the monitor which displays the desired information. Hence, utility subroutines to identify frame numbers would be desirable, but not critical to the total design.

The major tasks to be performed by the software, therefore, must match with the needs of the design of the videodisc modules. According to the designers/programmers at the Interactive Videodisc Research Laboratory, the essential tasks for the videodisc are as follows:

1. Jump to any frame number on the disc and display that image on the monitor.
2. Play forward/reverse at a normal speed of 30 frames per second.
3. Stop the disc from playing forward/reverse and display the current image continuously.
4. Stop the disc from playing and blank out the video image.
5. Play a particular motion sequence starting at a given frame and playing continuously- with audio- until a desired frame is reached.
6. Interrupt a motion sequence before it reaches its stop frame and allow the program to perform another task.
7. Turn the display, audio channels, etc. on or off as needed.
8. Insert all questions, branching, and graphics as desired.

After careful review, the following functions were added:

9. Clear the videodisc player from any error condition.
10. Step the player sequentially forward/reverse by one frame increments.
11. Turn the videodisc motor on or off to allow a trainee to turn the videodisc from side one to side two.
12. Scan forward/reverse.

The desired modules for the videodisc interface must allow performance of the above tasks as easily as possible and in any logical order.

HIERARCHICAL DESIGN OF THE VIDEODISC INTERFACE

The design to accomplish these tasks is divided into a hierarchy of three distinct levels. The top level, called the Application Level or Level 1, is the code or sequence written by a designer/programmer to produce a given videodisc package. Logically, this application level, has a start procedure, linear and looping structures, and an end routine. Below this level is the Videodisc Interface Level or Level 2. This level contains all the subroutines needed to access the videodisc player. Below the level is the Videodisc Utility Level or Level 3 which facilitates the details necessary to achieve the desired functions at Level 2. Level 3 basically controls the asychronous communications buffer for the videodisc.

BEST COPY AVAILABLE

Table 1 pictures the three levels.

TABLE 1

The Three Levels of the System Design for the Videodisc Interface

	MODULE NAMES
Level 1 Application Level	START USER-DEFINED END
Level 2 Videodisc Interface Level	START ERROR SEND FUNDAMENTAL COMMAND SEARCH FOR START FRAME PLAY UNTIL STOP FRAME
Level 3 Videodisc Utility Level	SEND AND CHECK COMPLETE CLEAN BUFFER

Level 2, the Videodisc Interface Level, consists minimally of five separate modules. The module START will do all housekeeping details to establish initial communications with the videodisc player. A second module, ERROR, will define the events (at any level) in case of an error condition. This module, moreover, must give the programmer as much information as possible in order to correct an inherent error condition. After displaying this information, the program will end abnormally. The next module is a basic routine for sending fundamental commands to the player. Fundamental commands are defined as those videodisc commands which are somewhat parallel in construction and require little or no formatting or alteration. "Turn audio on/off" or "clear" the discplayer are considered fundamental commands. The next two modules, SEARCH FOR START FRAME and PLAY UNTIL STOP FRAME, are considered to be subroutines that implement extended videodisc commands, those commands which require more formatting and special checking procedures in real-time. Except for the existence of an error procedure, the designer/programmer's hierarchical interface is represented in Figure 2.

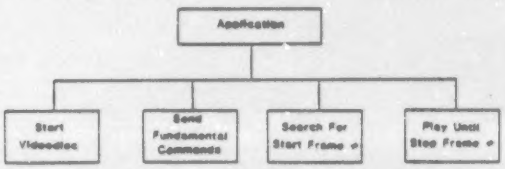


Figure 2. Programmer's View of the Videodisc Interface.

Flowcharts, moreover, are given in Figure 3, Figure 4, Figure 5 and Figure 6 for START, SEND FUNDAMENTAL COMMAND, SEARCH FOR START FRAME, and PLAY UNTIL STOP FRAME respectively. Each flowchart is structured, that is, each component is one of three types - linear, branching or looping and each component has one entry point and one exit point.

The bottom layer, the Videodisc Utility Level, moreover, contains the routines to make LEVEL 2 work. Generally speaking, the designer/programmer does not have to have any knowledge of this level in order to program for the interactive videodisc package. From Table 1, the module named SEND AND CHECK sends any command, either fundamental or extended, to the videodisc player and checks for the proper acknowledgement. The procedure called COMPLETE, meaning check for completion in real-time of an extended command, verifies the proper return code after a search for a start frame or play until a stop frame. The module entitled, CLEAN BUFFER, is an additional check to ensure that communications are clear before program control returns to upper levels. Thus, Level 3 does the work for Level 2 through three subroutines SEND AND CHECK, COMPLETE, and CLEAN BUFFER.

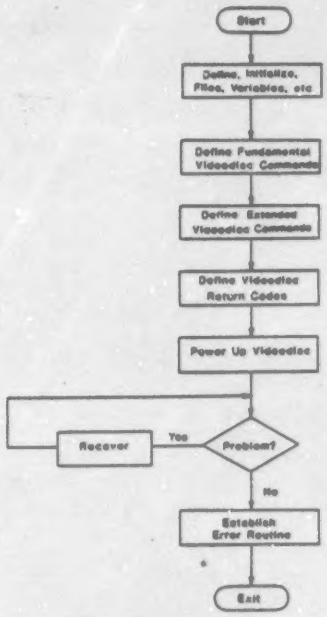


Figure 3. Flowchart for Subroutine START

In order to make the modules identifiable, each subroutine is prefixed with a "VS" or "VU". The "VS" symbol will identify a subroutine at Level 2 and "VU" will designate a routine at Level 3.These symbols will refer to Videodisc Subroutine

BEST COPY AVAILABLE

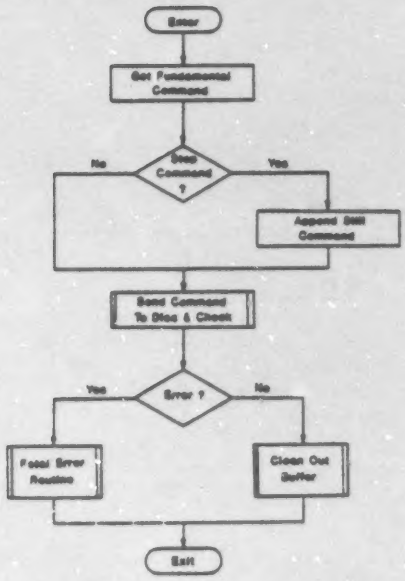


Figure 4. Flowchart for Subroutine SEND FUNDAMENTAL COMMAND

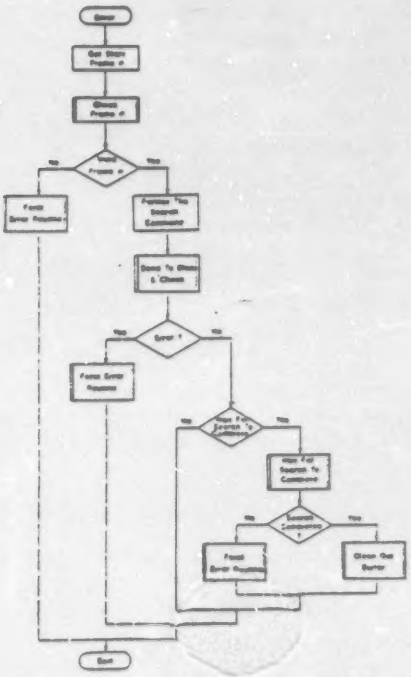


Figure 5. Flowchart for Subroutine SEARCH FOR START FRAME

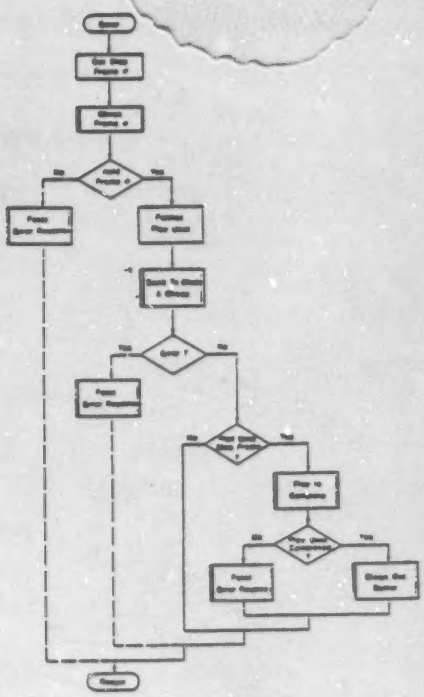


Figure 6. Flowchart for Subroutine PLAY UNTIL STOP FRAME

and Videodisc Utility respectively. The commands to the videodisc, moreover, and parameters used to connect the modules will all start with a "V". These conventions for naming subroutines and variables will make the design more readable and self-documenting. A detailed hierarchy chart showing the overall design with procedure names and levels is given in Figure 7. The numbers beside each subroutine are an additional identifier.

USER INTERFACE

The most important feature of the design is the correct use of reliable subroutines to drive the videodisc player. To achieve this objective, the input and/or output parameters for each subroutine, particularly those called from the Application Level, must be clearly specified. The VS.START housekeeping routine has no input or output parameters. The VS.ERROR routine is involked from within the VS.START procedure. Hence, a call to this routine is not necessary or desirable. Each of the remaining three modules at Level 2 requires a single input parameter. To utilize the subroutine VS.FUNDAMENTAL.COMMAND to send fundamental commands to the disc player, the parameter V.SEND must be loaded with the name of a fundamental command. (A list of fundamental commands

BEST COPY AVAILABLE

BEST COPY AVAILABLE

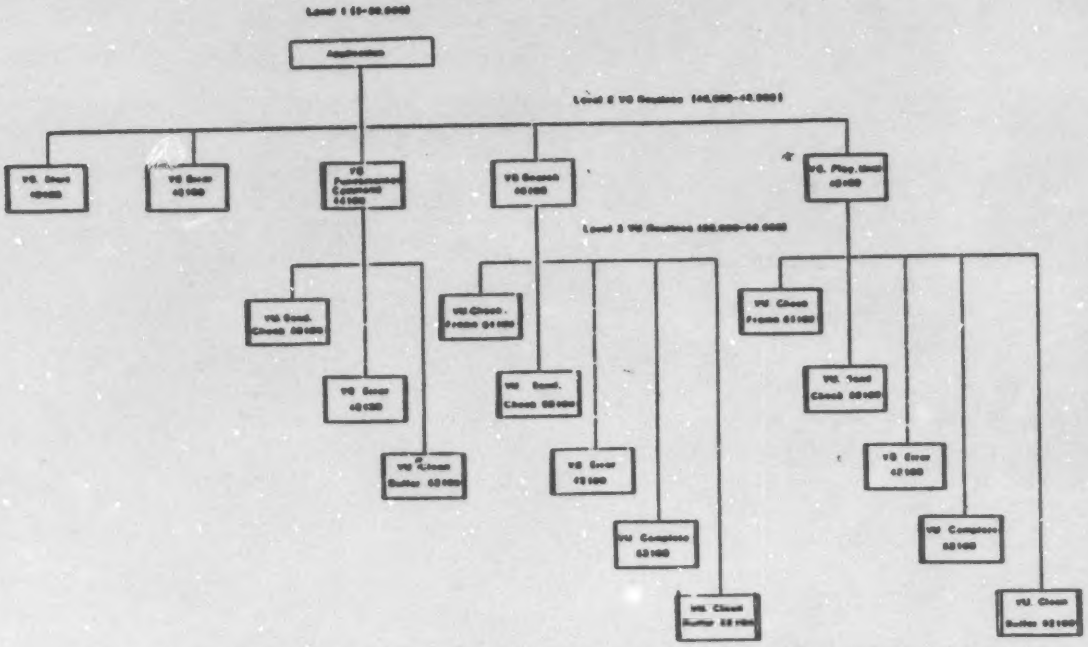


Figure 7. Detailed Hierarchy Chart for the Videodisc Interface.

is given in Table 2). For example, to activate the second audio channel to the ON state the following pseudocode is needed:

```
V.SEND = V.AUDIO2.ON
CALL VS.FUNDAMENTAL.COMMAND(V.SEND)
```

The control code of V.AUDIO2.ON is the input parameter utilized by the routine VS.FUNDAMENTAL.COMMAND. To utilize the routine VS.SEARCH, the input parameter, called V.START.FRAME, is given a one to five character value which represents a valid frame number on the videodisc. Likewise, the input parameter for VS.PLAY.UNTIL is a valid character value for the parameter entitled V.STOP.FRAME. All three routines have a common output parameter of V.FLAG. This flag, if activated, signals a user defined error condition. User defined errors could be inappropriate reply from the videodisc or an invalid frame number. The input parameters to the subroutines in the Videodisc Interface Level from the Application Level are given in Figure 8. An arrow with an open circle indicates data and an arrow with a shaded circle depicts control information. The direction of the arrow indicates whether it is an input or output to the subroutine.

Examples are appropriate. Assuming the procedures used in Figure 8, the user interface is simple. In order to play the displayer, the pseudocode is:

```
V.SEND=V.PLAY
CALL VS.FUNDAMENTAL.COMMAND(V.SEND)
```

171

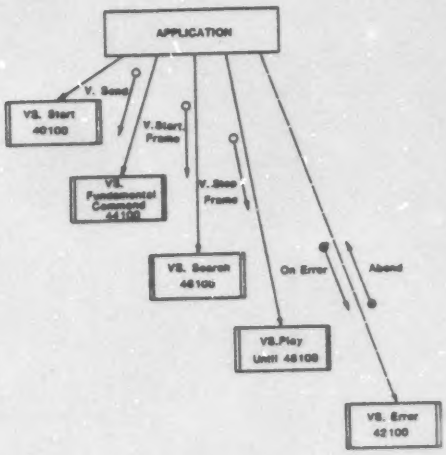


Figure 8. Structure Chart for the Application Level.

The pseudocode to play the disc from frame 1368 to frame 3674 (assuming the player is not on frame 1368), is as follows:

```
V.START.FRAME=1368
CALL VS.SEARCH(V.START.FRAME)
V.STOP.FRAME=3674
CALL VS.PLAY.UNTIL(V.STOP.FRAME)
```

To step still the displayer 20 times, use the following:

```
V.SEND=V.STEP.STILL
FOR J= 1 TO 20
  CALL V.FUNDAMENTAL.COMMAND(V.SEND)
END FOR
```

BEST COPY AVAILABLE

TABLE 2.
FUNDAMENTAL COMMANDS FOR THE VIDEODISC PLAYER

Program Name	Explanation
V.AUDIO 1	Toggle for audio 1 (CH 1)
V.AUDIO 1.OFF	Turn audio 1 OFF
V.AUDIO 1.ON	Turn audio 1 ON
V.AUDIO 2	Toggle for audio 2(CH2)
V.AUDIO 2.OFF	Turn audio 2 OFF
V.AUDIO 2.ON	Turn audio 2 ON
V.C.E.	Clear last entry or char sent
V.CLEAR	Clear last command
V.CONTINUE	Continue playing
V.DISPLAY	Toggle for displaying frame numbers
V.DISPLAY.OFF	Turn frame# display OFF
V.DISPLAY.ON	Turn frame# display ON
V.FAST.F	Play forward at 3x20 frames per second
V.FAST.R	Play backwards at 3x30 frames per second
V.MOTOR.OFF	Turn videodisc motor OFF
V.MOTOR.ON	Turn videodisc motor ON
V.PLAY	Play forward at 30 frames per second
V.PLAY.R	Play reverse at 30 frames per second
V.SCAN.F	Scan forward at 1/3 normal speed
V.SCAN.R	Scan backward at 1/3 normal speed
V.STEP.F	Step one frame forward
V.STEP.R	Step one frame back
V.STILL	Stop on current frame # with picture
V.STOP	Stop on current frame # with no picture

A similar explanation could be given for each subroutine in Level 3, the videodisc utility layer. The information for explaining the input and output parameters used by these subroutines is given in Figure 9, Figure 10 and Figure 11. Knowledge of the parameters between Level 2 and Level 3 is not needed by the programmer responsible for the Application or topmost level. Hence, the subroutines are readable, usable, and documentable for use by the programmer without any technical knowledge of the workings of the videodisc.

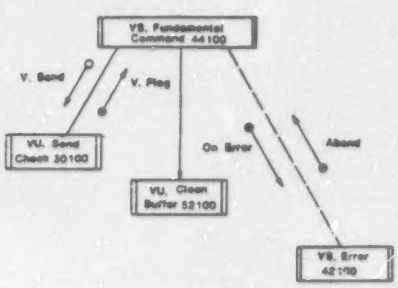


Figure 9. Structure Chart for VS.FUNDAMENTAL.COMMAND

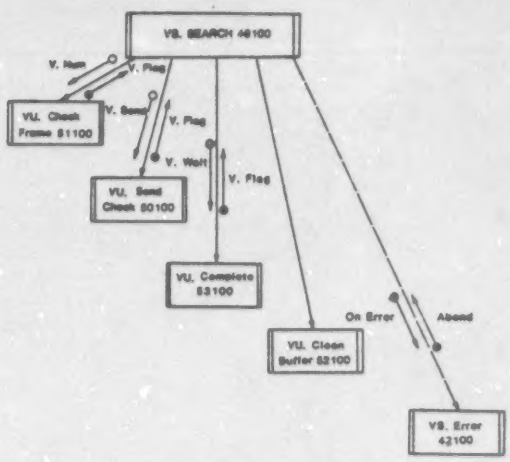


Figure 10. Structure Chart for VS.SEARCH

SOFTWARE ENHANCEMENT

Normally, the VS.SEARCH and VS.PLAY subroutines do not transfer program control to the Application Level until subroutines have checked for successful completion of the action required. This state of input/output can be changed to asychronous input/output. If it is desirable to perform another task before the completion is finished, a V.WAIT switch can be utilized. If this switch has the

BEST COPY AVAILABLE

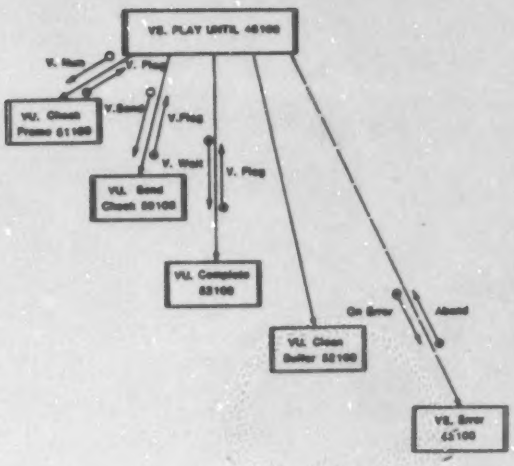


Figure 11. Structure Chart for VS.PLAY UNTIL

value "NO", then control will pass to the application level without checking for completion. (See the wait decision on the structure flowcharts for SEARCH and PLAY UNTIL subroutines in Figure 5 and Figure 6). This feature is included anticipating the need for overlay graphics while the search is being performed or to let a participant interrupt or stop a motion sequence. If this feature is incorporated into the design, the code at the Applica- Level is more complicated. The task of controlling and keeping the input buffer synchronized with the expected results from the videodisc player is then transferred to Level 1.

Two level 2 procedures would then have an additional input variable of V.WAIT. If it is desirable in the instructional design to wait until a Search or Play Until is completed, the following would be appropriate:

```
V.WAIT = YES
CALL V.SEARCH (V.START.FRAME,V.WAIT)
 OR
V.WAIT = YES
CALL V.PLAY.UNTIL(V.STOP.FRAM,V.WAIT)
```

If the design calls for the interrupting of a motion sequence, this pseudocode is needed:

```
V.WAIT= NO
CALL V.PLAY.UNITL(V.STOP.FRAME,V.WAIT)
.
.
.
Check for completion.
```

The last statement, Check for Completion, means that the task of checking for completion is done by the Application Level. The top level is now more complicated, but asychronous communication between devices is achieved. From the participants' point of view this infers that he can interrupt a motion squence to review the first part, stop the entire sequence or branch to any other state.

CONCLUDING REMARKS

The design of this software interface was implemented at the Interactive Videodisc Research Laboratory during the Summer of 1984. The personal computer of the system is an IBM PC using PC DOS 2.1, the intelligent videodisc player is a SONY LDP-1000, and the language is BASICA,VA(2.1). Serial ports (RS232C) exist on both the computer and videodisc player. The subroutines were used, with asychronous input/output with bar code (Baur, 1984) and with touch sensitive screen (Miller,1984).

The purpose of this software interface is to give the producer of videodisc courseware a flexible design needed to combine various media. The interested designer and/or programmer is encouraged to enhance, modify, and alter these modules to fit the requirements of his videodisc system.

REFERENCES

Baur, G.R. The use of bar code input/output in an intelligent interactive videodisc instructional system,MSEE Thesis, University of Notre Dame, August, 1984.

Daynes, R. The videodisc interfacing primer, BYTE, June 1982, pp. 48-59.

Graham, N. Introduction to computer science, New York: West Publishing Company, 1982.

Miller, T. Videodisc Premastering and Production Techniques for an Integrated and Interactive Microcomputer-based Learning System, MSEE Thesis, University of Notre Dame, September, 1984.

Pigford, V.D., Henry, E.W. and Miller, A.E., Assessing integrated and interactive microcomputer based videodisc authoring systems, Paper presented at the Second Annual IEEE Ed Comp Con, San Jose, November 1984.

Pigford, V.D., Interactive videodisc: a model for authoring systems and a software interface for microcomputers, MSEE Thesis, University of Notre Dame, August 1984.

Weinberg, V.,Structured analysis, New York: Yourdon Press, 1979.

BEST COPY AVAILABLE

AN INTERACTIVE VIDEO CURRICULUM MODEL

Mark Sanders - Virginia Polytechnic Institute & State University

ABSTRACT

Interactive video (I/V), the marriage of microcomputing and video technologies, is at the vanguard of a host of new instructional technologies. While this new educational medium is rapidly gaining advocates in the business, industry, and military sectors, its introduction to the educational community is still very much in the formative stage. After working on and off for a year designing and developing interactive video instructional materials, the author constructed and field tested an interactive video curriculum with college students enrolled in a teacher education program. This paper provides a synopsis of that curriculum model, with recommendations to those interested in teaching others the process of developing interactive video instructional materials.

FIELD TEST CONDITIONS

The interactive video curriculum was field tested with twelve students enrolled in a course regularly taught by the author entitled "Special Problems in Graphic Communications." This course is among the offerings of the Industrial Arts Program Area in the College of Education at Virginia Polytechnic Institute and State University. The field test was conducted during the Spring, 1984 quarter.

Although the class met six hours per week for eleven weeks, the interactive video unit comprised only one of several major units taught during the course. Ten of the twelve students had no prior instruction in the use of microcomputers. One complete workstation was in place in the graphic communications laboratory where the class met regularly. In addition, a dozen microcomputers were available to students six days a week in a separate "Self Instructional Curriculum Laboratory" maintained by the College of Education.

Each student was assigned the task of designing and developing an interactive video instructional program using a conventionally produced videotape which the author provided. The tape contained eight short demonstrations of graphic arts (printing) processes ranging from about three to ten minutes in length. Each student was given the opportunity to select one of these video demonstrations as the basis for their project. The completed I/V programs were required to include one or more graphic pages with overlayed text produced by a character generation program. The length of the I/V program, quantity of programed interaction, and overall design were left up to the discretion of each student.

HARDWARE/SOFTWARE REQUIREMENTS

The interactive video system was built around a 64K Apple II Plus microcomputer that was in use in the graphic communications laboratory. Additional hardware included one 5 1/4" disk drive, a color monitor, a Sony SLO-325 video cassette recorder (VCR), an I/V interface card manufactured by BCD Associates, and a cable to connect the VCR with the interface card (See Figure 1). A dot matrix printer, though technically optional, was instrumental during the development of the I/V programs, allowing students to edit lessons in hard copy before making corrections in their programs.

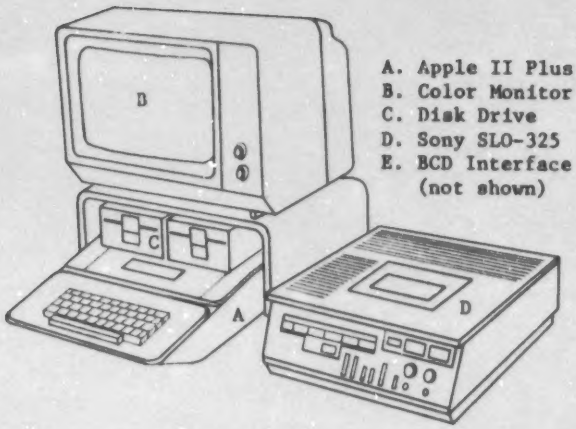


FIGURE 1

The BCD interface card was selected from among several available cards because it was the least expensive Apple compatible card that provided absolute frame addressability. That is, the "in" and "out" points of each video scene could be identified to the nearest video frame (1/30 second). This was essential, particularly in retroactively converting a conventional videotape into an interactive video lesson. The Sony VCR was one of several compatible "industrial" VCR's. Consumer VCR's were not compatible with the I/V interface card, nor were they considered durable enough for classroom use.

To produce the graphics for the I/V courseware, the Koala Pad, an inexpensive digitizing tablet manufactured by Koala Technologies, was employed. This enabled inexperienced students to rapidly generate sophisticated graphics that could be called up as needed by the I/V courseware they wrote.

The primary software required for the I/V curriculum was the BCD course authoring software called "The Instructor" (BCD Associates, 1983). This software includes both a lesson authoring program and a tape logging program. The former allows the instructional developer to write the I/V lesson, including information pages, question pages, and branching instructions. The logging program provides the mechanism for writing a special time code on the videotape and then identifying start and stop points for individual video scenes within the conventional videotape.

"Micro Illustrator" (Dompier, 1983) was the software used with the Koala Pad to generate graphic pages. "Apple Mechanic" (Kersey, 1982) provided the character generation program that students used to add titles/captions to their graphic pages. In addition, "DOS 3.3 System Master" programs were used to handle routine utility functions such as copying disks and files.

Table 1

Hardware/Software Requirements

Hardware	Cost
Apple II Plus/IIe	$1050 (Educational Price)
5 1/4" Disk Drive	
Color Monitor	
Dot Matrix Printer	$300
Sony SLO-325 VCR	$1350
I/V Interface Card (BCD Associates)	$600
Koala Pad (Koala Technologies)	$85
Software	**Cost**
"The Instructor" (BCD Associates)	$395
Micro Illustrator (Koala Technologies)	Included
Apple Mechanic (Beagle Brothers)	$30
DOS 3.3	Included
Total	$3810

TEXT/READINGS

While there are currently several interactive video textbooks on the market (See, for example, Floyd and Floyd, 1982), this curriculum model is supported by a collection of readings and software documentation. Readings were placed on reserve in the Self Instructional Curriculum Laboratory, and were therefore available to students as they worked on their programs outside of class. Software documentation and appropriate references included documentation for "The Instructor," "Apple Mechanic," and "Micro Illustrator," as well as The DOS Manual (Apple Computer, Inc., 1981). These provided students with ready references while learning to use the software. In addition to these learning aids, The Systematic Design of Instructor (Dick and Carey, 1978) provided a very logical and useful description of the instructional

design process.

INTERACTIVE VIDEO COURSE OUTLINE

The following interactive video course outline is basically that used with students in the field test. Some modifications have been made based upon the results of the field test. The course is designed to teach the I/V development process to microcomputer novices at the postsecondary level.

I. Introduction
  A. Overview of I/V
  B. Foundations of I/V
    1) Programed Instruction
    2) Video Instruction
    3) Computer Assisted Instruction
  C. Student Assisted Instruction

II. Hardware
  A. Introduction to Microcomputers
  B. Introduction to Video Cassette Recorders
  C. Introduction to Peripheral Hardware

III. Software Requirements for I/V
  A. Video Logging Software
  B. Course Authoring Software
  C. A Word About BASIC

IV. Instructional Design Process (Dick and Carey)
  A. Identifying Instructional Goals
  B. Instructional Analysis
  C. Entry Behaviors
  D. Performance Objectives
  E. Criterion Referenced Tests
  F. Instructional Strategy

V. DOS Understandings
  A. Commands
  B. Utility Programs

VI. Lesson Development (The Instructor)
  A. Scripting
  B. Writing Information Pages
  C. Writing Question Pages
  D. Writing Graphic Pages
  E. Understanding Branching

VII. Video Logging
  A. Identifying Video Scenes
  B. Logging Video Scenes (The Instructor)

VIII. Preparing Graphics
  A. Digitized Graphics (Koala Pad)
  B. Programed Graphics (Applesoft BASIC)

IX. Character Generation
  A. Character Generation Software (Apple Mechanic)
  B. Overlaying Characters on Graphics

X. Preparing an I/V Courseware Disk

XI. Evaluating I/V Courseware

CONCLUSIONS AND RECOMMENDATIONS

The field test proved to be a very positive experience for both the author and students. Students responded enthusiastically to the assignment. The task was manageable, the video element held their interest, and for most it was a successful introduction to the vast potential of microcomputers. On balance, this instructional unit went rather smoothly, with each of the students completing an I/V program by the end of the quarter.

The eleven week quarter was more than adequate time for program development. With other assignments required concurrently, most students spent less than half the quarter on the I/V component. Although one complete workstation in the graphic communications laboratory provided students enough time to view and log their video scenes, one of more additional workstations would certainly have been well utilized. The arrangement with the Self Instructional Curriculum Laboratory that allowed students to work on Apples after class really worked out nicely. With the exception of the video logging, all of the work could be completed at this facility, thereby freeing up the workstation in the classroom for video work only.

Teaching the interactive video development process turned out to be a good way of introducing microcomputer fundamentals to the uninitiated. The I/V assignment provided these would-be educators with a relevant "state-of-the-art" microcomputer application. In completing the assignment, students found they needed to learn basic file management, DOS commands, and in some cases, a bit of programming. Thus, the process of becoming "computer literate" proved to be more a matter of problem solving than an academic exercise.

It is essential that students work from an existing videotape rather than attempt to develop their own for this purpose. Video production is to most time consuming and costly task in the I/V developmental process. While any videotape may be logged and therefore retroactively converted to interactivity, only short video segments are practical for those learning the process. Adding length to the task increases the development time, but not necessarily the learning potential.

Interactive video may be thought of as CAI that utilizes video as a part of the delivery system. If you remove the video component, students are left with a solid CAI learning experience. This model could in fact easily be adapted to do just

that. While the interactive video development process may be viewed by many as a very costly and time consuming endeavor, the curriculum model here provides an efficient alternative for those interested in getting started in the field.

REFERENCES

Dick, Walter & Carey, Lou. The systematic design of instruction. Glenview, IL: Scott, Foresman and Company, 1978.

Dompier, Steven. Micro illustrator. Santa Clara, CA: Koala Technologies, Inc., 1983.

The DOS manual. Cupertino, CA: Apple Computer, Inc., 1981.

Floyd, Stephen & Floyd, Beth (Eds.), Handbook of interactive video. White Plains, NY: Knowledge Industry Publications, Inc., 1982.

The instructor. Oklahoma City: BCD Associates, Inc., 1983.

Kersey, Bert. Apple mechanic. San Diego: Beagle Brothers Micro Software Inc., 1982.

Computer Aided Instruction of Descriptive Geometry

Lester F. Smith
The Ohio State University
Columbus, Ohio

Mark R. Hopkins
OMNICAD
Fairport, New York

Jacob J. Wolf III
University of Pittsburgh
Pittsburgh, Pennsylvania

Ding T. Jou
IBM Corporation
New Paltz, New York

ABSTRACT

The Ohio State University has received a grant from The International Business Machine Corporation (IBM) to develop software programs for instruction in descriptive geometry.

Descriptive Geometry, that segment of graphic science which determines the relationship between points, lines and surfaces, traditionally has been practiced with T-squares, triangles, protractors, compass etc.

An Interactive Computer Program, DEGE, was written to replace the traditional "tools" with a two screen interactive workstation.

The "Interactive Workstation", assembled and used by the Interactive Graphics Laboratory of the Ohio State University, contains the following components:

1. IBM 3277 Display Station
2. Tektronix Graphics Monitor-618
3. Cursor Control Unit
4. Tektronix 4611 Hard Copy Unit
5. IBM 4341 Computer

The Interactive Graphics Laboratory is contained within the Department of Engineering Graphics. DEGE was evaluated as an instructional tool by using DEGE in the second term of the 1984 Summer Quarter (7/29/84-9/3/84). DEGE was written to obtain:

1. Reduced time demands on students,
2. Increased student understanding,
3. Improved accuracy of drawing,
4. Improved clarity of drawing,
5. Reversible recovery capability,
6. Exposure of students to computer graphics.

The effectiveness of DEGE in obtaining the objectives will be discussed.

Introduction

The Engineering Graphics Department of The Ohio State University has been developing computer programs to assist students in acquiring the graphical skills necessary to function as engineers. Computer programs have been written to:

1. Tutor the students on selected topics
2. Teach concepts of isometric drawing
3. Teach concepts of orthographic projection
4. Teach concepts of descriptive geometry
5. Teach concepts of dimensioning
6. Teach concepts of flowcharting
7. Teach concepts of solid body rotation

Several of the computer programs have progressed to the point of sharing our findings with the educational community. DEGE, a computer program to teach the concepts of descriptive geometry, will be the focus of this paper.

Engineers must know the relationship between points, lines and surfaces of a proposed design. Designs almost universally, involve three dimensions and thus the orthographic view possessing the desired information must be drawn using the traditional tools. T-square, triangles, protractors, compasses etc. are associated with engineers who use them with varying degrees of efficiency. Teaching engineering undergraduates the concepts of descriptive geometry is difficult and assessing a students competency is even more difficult due to variations in the physical capabilities of students to do accurate work. Some students possessing the mental skills will create "Picasso's" while other students possessing lesser mental skills will create "Rembrandt's".

DEGE eliminates physical skills as a variable and thus permits the faculty member and the student to focus his/her

attention on understanding concepts. The objectives of DEGE are:
1. Reduced time demands on students,
2. Increased student understanding
3. Improved accuracy of drawing,
4. Improved clarity of drawing,
5. Reversible recovery capability,
6. Exposure to computer graphics.

The effectiveness of DEGE in accomplishing the objectives will be discussed in "Implementation and Results".

Software Development

The process of writing a computer program to solve descriptive geometry problems in a manner similar to traditional methods was facilitated by conceptualizing the process into four separate segments; namely,
1. Interpreting DEGE user input.
2. Obtaining an additional view of a point.
3. Obtaining an additional view of a curve.
4. Operational.

Each segment has a unique objective and employs subroutines to accomplish it. The objective of each segment, the subroutines used in each segment and the purpose of each subroutine are:

Segment 1 Interpreting User Input

Objective: To interpret the input of the user and to transform that input into a character string recognizable as a DEGE instruction.

Subroutines:

SEARCH - Scans user input string, determines beginning and end of string; establishes point arrays.

IDTYPE - Identifies the type of descriptive geometry entity - point, line, plane etc. - rand its location.

DECODE - Decides what was identified by IDTYPE.

Segment 2 Obtaining an Additional View of a Point

Objective: To create a point or series of points in a new view.

Subroutines:

DIRERP - Directs the two reference planes and obtains their equations.

FINDPT - Computes the coordinates of a point in the new view using the coordinates of the point in the view from which the new view is projected and the coordinates of the point in the view from which distances are transferred.

Segment 3 Obtaining an Additional View of a Curve

Objective: To create a curve in a new view.

Subroutines:

LINARC - Computes the X, Y values of the intersection points between a line and a curve.

GENARC - Generates the X, Y values of 48 points on a curve. The major and minor axes of an ellipse are the input parameters

Segment 4 Operational

Objective: To perform specific operations, such as drawing a point, line, view, circle, changing a graphic element, deleting a graphic element, redisplaying the graphics screen, indicating errors and selecting a problem from data bank.

Subroutines:

POINT - Locates and names the points of intersection of two graphic elements.

LINE - Draws and labels a line.

VIEW - Draws an additional view of a graphic object.

CIRCLE - Draws a circle in a view where the circle appears as a circle or as a line.

CHANGE - Changes the image of an element on the graphics screen.

DELETE - Deletes a graphic element from memory and the graphics screen.

REDISPLAY - Redisplays the graphics screen with/ without support details.

ERROR - Displays on alphanumeric screen the error messages and hints for corrective action.

SELECT - Data bank contains descriptive geometry problems used in Engineering Graphics 141/142/143. User may select a problem to solve using DEGE.

Interface: User-DEGE

The user must communicate with DEGE by using "COMMANDS" or "SPECIFICATIONS". DEGE prompts the user, via the alphanumeric screen, with a request for a problem selection from the data bank. The user, upon selecting a problem, will see "COMMAND" on the alphanumeric screen and the descriptive geometry problem on the graphics screen. DEGE accepts "specification" only in response to a DEGE prompt

for a "SPECIFICATION" which occur within a "COMMAND".

DEGE commands and the associated specifications are:

<COMMAND> LINE
Function: To draw a line.

SPECIFICATIONS
CONNECT <POINT> TO <POINT>
PARALLEL <LINE> OFFSET <NUMBER>
PA <LINE> THROUGH <POINT>
PERPENDICULAR <LINE> TH <POINT>
CROSS <LINE> TH <POINT> AT <NUMBER>
DIVIDE <LINE> INTO <NUMBER>
DI <ANGLE> IN <NUMBER>
TANGENT <CURVE> AND <CURVE>
TA <CURVE> TH <POINT>
TA <CURVE> PA <LINE>
TA <CURVE> PE <LINE>

<COMMAND> POINT
Function: To name the point(s) of intersection of two elements.

SPECIFICATIONS
LINE LINE
CURVE CURVE

<COMMAND> VIEW
Function: To draw a view.

SPECIFICATION
VIEW
POINT
LINE VIEW
OBJECT
CURVE

In some instances the line of sight must be identified by specifying a direction e.g. PA AB/T.

<COMMAND> CIRCLE
Function: To draw a circle.

SPECIFICATION
CENTER <POINT> RADIUS <NUMBER>
CENTER <POINT> THROUGH <POINT>
CENTER <POINT> TANGENT <LINE>
THROUGH <OBJECT>
INSCRIBE <OBJECT>
EDGEVIEW <CURVE> ON <LINE>

<COMMAND> CHANGE
Function: To change the position of a label for a point or to change the line type for a line or a curve.

SPECIFICATION
Specifications for CHANGE, REDISPLAY, DELETE are entered by the user as an addendum when replying to the <COMMAND> prompt. DEGE draws all lines as construction lines and labels all points in the 1st position.

Line Type Correlation

Number	Appearance	Significance
0		Invisible Line
1	————	Construction Line
2	————	Visible Object Line
3	- - - - - -	Hidden Object Line
4	—— - ——	Center Line

Label Position

Number	Position
1	A_T
2	A_T +
3	A_T +
4	+ A_T
0	+

CHANGE AB/T TO 4 will change line AB in the front view to a center line.

<COMMAND> DELETE
Function: To delete a point, line, object, curve or an entire view.

DE AB/T will cause points A, B and the Line AB in the top view to be deleted as well as any line containing points A and B.

<COMMAND> REDISPLAY
Function: To redisplay the graphics screen.

RE 1 2 3 permits the user to select 1 of 3 levels of support detail

<COMMAND> COPY
Function: To produce a paper copy of the graphics screen.

<COMMAND> QUIT
Function: To terminate a graphics screen.

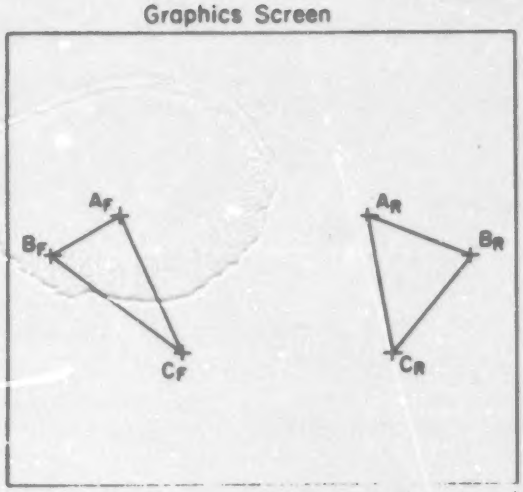
A sequential depiction of the alpha-numeric screen and a selected depiction of the graphics screen illustrates the usage of commands and specification to solve a problem.

<COMMAND> SELECT User commands are enclosed in boxes. System responses are noted by *** ***

<ENTER THE NAME OF THE DESIRED PROBLEM>

4.62

Graphics Screen



Problem 4.62 Show the front view of the circle that passes through points A, B and C.

<COMMAND>
LINE
*** SPECIFY HOW TO DRAW THIS LINE ***

PERPENDICULAR A/FR THROUGH C/R

<COMMAND>
POINT
*** SPECIFY THE TWO ELEMENTS TO BE INTERSECTED ***
$1 AB/R
*** HIT A POINT-NAME-KEY (1 TO 9 OR A TO Z) TO NAME ***

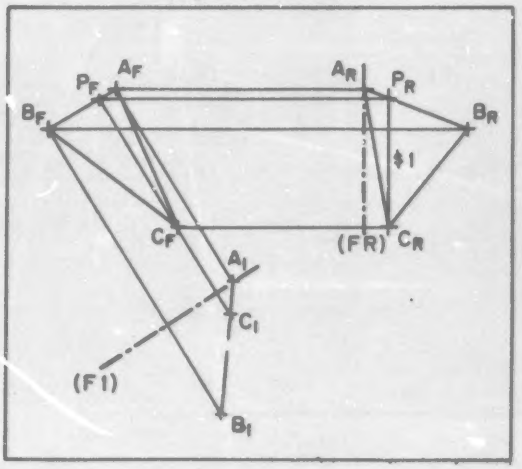
<COMMAND>
POINT
*** SPECIFY THE TWO ELEMENTS TO BE INTERSECTED ***
P/FR AB/F

<COMMAND>
LINE
*** SPECIFY HOW TO DRAW THIS LINE ***
CONNECT P/F TO C/F

<COMMAND>
VIEW
*** SPECIFY THE OBJECT TO BE PROJECTED AND THE NEW VIEW ***
ABC/F 1
*** ESTABLISH THE LINE OF SIGHT ***
PARALLEL CP/F
*** INDICATE THE VIEW FROM WHICH THE DISTANCES ARE TRANSFERRED ***
R
*** LOCATE THE REFERENCE PLANE BETWEEN VIEW-F AND VIEW-1 BY OPERATING JOYSTICK ***
*** HIT F-KEY TO ENTER THE FINAL POSITION ***

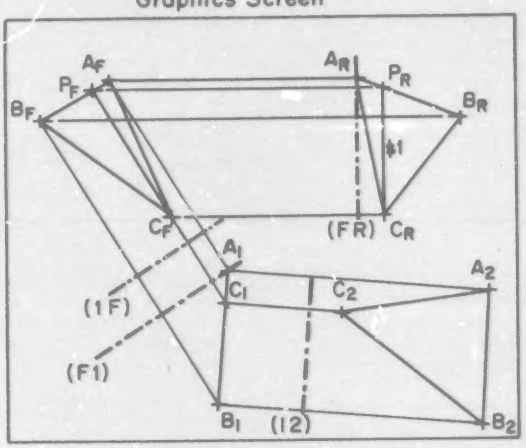
*** LOCATE THE REFERENCE PLANE BETWEEN VIEW-F AND VIEW-R BY OPERATING JOYSTICK ***
*** HIT F-KEY TO ENTER FINAL POSITION ***

Graphics Screen



<COMMAND>
VIEW
*** SPECIFY THE OBJECT TO BE PROJECTED AND THE NEW VIEW ***
1 2
*** ESTABLISH THE LINE OF SIGHT ***
PERPENDICULAR AB/1
*** INDICATE THE VIEW FROM WHICH THE DISTANCES AND TRANSFERRED ***
F
*** LOCATE THE REFERENCE PLANE BETWEEN VIEW-1 AND VIEW-2 BY
OPERATING JOYSTICK ***
*** HIT F-KEY TO ENTER THE FINAL POSITION ***
*** LOCATE THE REFERENCE PLANE BETWEEN VIEW-1 AND VIEW-F BY
OPERATING JOYSTICK ***
*** HIT F-KEY TO ENTER THE FINAL POSITION ***

Graphics Screen



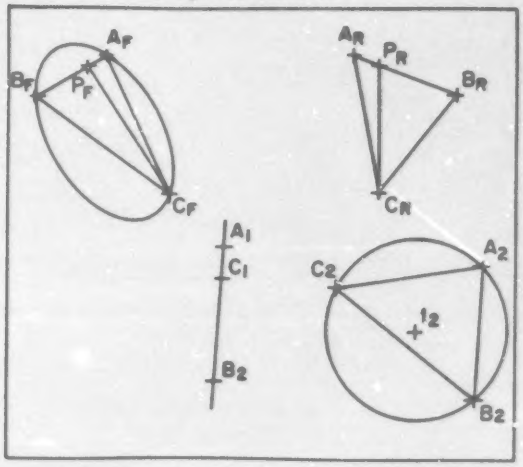
```
<COMMAND>
   CIRCLE
*** SPECIFY HOW TO DRAW THIS CIRCLE
   THROUGH ABC/2

<COMMAND>
   CIRCLE
*** SPECIFY HOW TO DRAW THIS CIRCLE ***
   EDGEVIEW @1 ON AB/1

<COMMAND>
   VIEW
*** SPECIFY THE OBJECT TO BE PROJECTED AND
THE NEW VIEW ***
   @2 F
*** INDICATE THE VIEW FROM WHICH THE
DISTANCE ARE TRANSFERRED ***
   2

<COMMAND>
   REDISPLAY 1
```

Graphics Screen



Implementation and Results

Engineering Graphics 142 at The Ohio State University has a significant component dedicated to instruction of descriptive geometry. Interactive computing using graphical concepts comprises the balance. DEGE was conceived as a tool to emcompass both topics effectively and economically. Twelve freshmen students were enrolled during the summer quarter and were required to solve 30 descriptive geometry problems - 20 using traditional methods and 10 using DEGE. Mr. Mark S. Stephenoff, GTA, taught the course. He is an experienced instructor and possessed a working knowledge of DEGE. Mr. Stephenoff critiqued a DEGE Student Manual which was written specifically for entering engineering freshman. The "Student Manual" contained numerous examples to graphically depict the written material. Assigned to students to solve on DEGE were problems in the following topics:

1. Primary and Secondary Views
2. View in Specified Direction
3. Edge and Normal Views of Planes
4. Angle Between Planes
5. Parallel Lines and Planes
6. Parallel Planes
7. Shortest Distance between Lines
8. Perpendicular Lines and Planes
9. Cylinders
10. Cones

The problems were selected by the instructor to be equal in time and difficulty with other problems to be solved using traditional methods. Two undergraduate student were employed as consultants to answer DEGE questions arising in the Interactive Graphics Laboratory. At the conclusion of the term (5 weeks) course the students were asked to complete a questionnaire. Nine students completod the questionnaire with the results being:

8/9 stated less time with an average time saving of 21%.
9/11 concepts better understood.
4 stated that the "Student Manual" adequately explained how to use DEGE to solve the problems of Descriptive Geometry - 4 stated it did not.
9/9 stated - DEGE was a pleasant experience.
7/9 They would recommend more EG 142 problems be solved on DEGE with the average being 15 problems (50% increase).
9/9 stated they would recommend DEGE to friends.

Responses to "How the manual could be improved?":
1. More examples,
2. Description of error removal techniques,
3. Explain in greater detail, how to use commands.

Responses to "What are the strong points of DEGE?":
1. Drawing on computer.
2. Drawing errors are reduced.
3. No erasers.
4. Time saved.

Responses to "What aspects of DEGE could be improved?":
1. Logging On/Off
2. Reference Plane Removal
3. "Equipment wasn't up to par"
4. Guided tour W/Introductory session.

Assessment of DEGE

Improvements are needed in the following areas:
1. Students Manual
2. Data base must be enlarged to include specific examples illustrating

concepts.
3. Simplify cylinder and curve procedures.
4. Amend DEGE to administer tests.
5. Amend DEGE to delete reference planes.

DEGE , a computer program with over 7000 FORTRAN statements was successful in reducing the time committment of students and in increasing student understanding of concepts and in motivating students - they enjoyed it!

Total acceptance of DEGE for all EG 142 students is severely limited by the capital investment in hardware. Exploratory discussions concerning the use of DEGE on personal computers were initiated in April 84 and are presently in the gestation period. The author's personal opinion is that DEGE must be delivered on hardware of reasonable cost with wide acceptance by the educational community. We are working toward that end.

Evaluation of Student Performance Using the Apple Microcomputer System

John H. Dustman

Indiana University Northwest

ABSTRACT

The measurement of student performance often represents a substantial, time-consuming task for the classroom teacher. The use of microcomputer technology and the development of a specific software system called CAS or Classroom Administration System has allowed the refinement of measurement techniques which otherwise would have been considered too time consuming or technically impossible just a few years ago. We are currently investigating the differences between measuring and reporting student performance in a test/grade environment versus a concept/performance environment. Student reaction to the use of the two different protocols of grade maintainence are discussed. The CAS system will be discussed in terms of its capabilities and limitations. This system has allowed full norm-referencing and criterion referencing of all testing within a college level Anatomy and Physiology course but can be equally applied to other courses at both the college and high school levels. Grade analysis and reporting techniques are discussed and related to the possible comparisons of scores over a multiyear period and with comparisons concentrating of instructor performance difference.

CAS COMPONENTS:

QUESTION DATA-BASE SYSTEM

ADD/EDIT QUESTIONS
PRINT QUESTIONS
KEY-TERM SEARCHES

for: MULTIPLE CHOICE
TRUE/FALSE
MATCHING
COMPLETION or FILL IN
ESSAY QUESTIONS

EXAMINATION PRODUCTION SYSTEM

includes: DIRECTIONS
MULTIPLE CHOICE QUESTIONS
MATCHING QUESTIONS
TRUE/FALSE QUESTIONS
COMPLETION QUESTIONS
ESSAY QUESTIONS

WORD PROCESSING OF TEXT
PRINTING OF EXAM KEY
PROJECTING GRADES BASED ON NORM REFERENCE DATA
SELECT QUESTIONS BY KEY TERM OR CODE SEARCHES

TEST SCORING & ANALYSIS SYSTEM

includes: MACHINE GRADING OF TESTS
FULL ITEM ANALYSIS
NORM-REFERENCE ANALYSIS
SUBSETTING OF TESTS
AUTOMATIC GRADEBOOK AND NORM-REFERENCE UPDATES

GRADEBOOK SYSTEM

includes: GRADEBOOK OF STUDENT SCORES
GRADE REPORTING BY PERCENT AND STANINES

Evaluation of Student Performance Using the Apple Microcomputer System

GRADE CURVING
GRADE ANALYSIS BY CODE FIELDS
INDIVIDUAL STUDENT REPORT
GENERATION

UTILITY SYSTEM:

includes: CATALOG AND COPY ROUTINES
CONVERSION PROGRAMS TO
OTHER SYSTEMS

HARDWARE REQUIREMENTS: Apple II, II+, IIe or IIc, or Apple III, IBM with Quadlink
One or Two disk drives
Mechanical card reader or scanner (optional)

One of the more time consuming aspects of teaching is the evaluation of student performance. The typical procedure for student evaluations involves the development and use of examinations covering the material included in the current course of study. The questions for each examination are selected to cover particular topics and/or objectives. As a result of the test, each student is traditionally awarded a numeric or letter grade which theoretically conveys, to both the student and the teacher, the progress that student has made in the course.

With the advent of computers and microcomputers, programs were rapidly developed to "aid" in this evaluative process by statistically analyzing the examination. The evaluative procedures which were used, however, were aimed at evaluating the test as a test and not the student as a test taker. In this egocentric approach, we consistently used such parameters as: Question Item Analysis, Kuder-Richardson Coefficients, Average, Mode, and Median, to express the characteristics and validity of the test as a test. Very little effort was expended to develop analytical evaluations of the student performance *per se*. To illustrate the limitations of this egocentric approach is the following example: An instructor gives an examination of 100 questions to 20 students. The typical "evaluation" will indicate that the test average was 75% and had a Kuder-Richardson of 0.81. It will also list how many students missed each question. From this evaluation we can decide if there are any questions which are overtly easy or difficult. The analysis will also show that Student "A" had a grade of 75% and Student "B" and "C" had grades of 75% for that examination. Based upon that information the logical conclusion would be that all three students performed equally well on the examination. However, this may not be the true interpetation of the rest data. Assuming that the test covered more than one topic, or had questions of different levels of complexity, these students could differ drastically by area or type of question missed. Few programs exist to make this type of judgement. Over the past 4 years we have developed and tested a program system called CAS (Classroom Administration System) which directly addresses the need for quick, efficient, and accurate evluations of both the examination and the student taking the examination.

CAS consists of 5 interdependent series of programs. There is a: (1) "Question Data-Base" System designed to generate and maintain question files; (2) "Examination Production" System designed to produce examinations from the question data-base; (3) "Test Scoring & Analysis" System to score examinations and produce various analytical reports; (4)"Gradebook" System to maintain student grades, produce grade reports and counseling information; and (5) "Utility" System to do a variety of support functions.

QUESTION DATA-BASE SYSTEM

The "Question Data-Base" system allows for the development of files of questions in either a "multiple Choice", "True/False" "Completion or Fill-in", "Matching" or "Essay" type format. Each question file, with the exception of the matching type, may have up to 49 questions per file, and up to 10 files per diskette. Each multiple choice question may be up to 230 characters in length, and each of 5 answer choices may be up to 160 characters each. Essay and Completion questions may be up to 325 characters in length. There may be up to 1000 Matching sets of terms per diskette, each with two 32 character terms.

The questions, with the exception of the matching type questions, are fully norm-referenced and may, in addition, be coded for course objective, performance criterion levels, type of question (Bloom Level I,II,II,orIV), or any other type of code the question designer might desire. The inclusion of Norm-referencing and Coding information allows for the analysis of the test data against a historical data base. Within the question information the necessary printer information can be stored to allow for sub- and super-scripts and underlining, for boldfacing and changing type styles. In the sciences, these features are important in composing test questions since, if the student is taught that the formula for water is H_2O and the test question asks what is H20, the lack of subscripting adds a new factor into the question.

In the development of the question data base system, a review of the possible protocols for storing data was made to determine which was most effective. Most programs that have been developed in this area have concentrated on storing data either as random-access records, or in sequential text files. The random access method being the primary choice because of the reduction of time involved in accessing the records. However, both methods seemed inappropriate for this application, and a new method of data storage was devised to greatly reduce disk storage space and disk access time. Table 1 compares the time and storage requirements of CAS as compared to the other methods of storing data.

TABLE 1:

Method of Storage	Storage of 50 Questions Read/Write Time	Sectors Needed
Random Access Length = 1030	3sec/record 150sec/50	210
Sequential Length 500/rec.	70 sec/50	100
CAS System	4sec/50	43

In those systems which use random access disk files a provision for the maximum record size must be made and this ultimately requires a substantial amount of disk storage area to be set aside for potentially blank parts of records, and increases the overall time to read an entire file. Sequential records allow for variable record sizes but requires more read/write time for individual records. In the CAS system only 4 seconds are to read 1 or 50 questions and only 43 sectors of disk storage space. This savings is accomplished by using a binary storage of a memory segment as a single record. Each time a file is stored or accessed, a 10K unit of memory is exchanged with the disk. Questions are stored in memory with flags separating parts of questions, and index variables for the starting and stopping locations for each question. This binary "snapshot" makes a very quick and efficient method of data exchange as compared to the usual methods of record storage, and allows a diskette to accomadate 500 questions rather than the 100 when a random access system is utilized. The only time when question files lose their efficiency is when the files only contain 1 or 2 questions.

The use of a memory snapshot to store and retrieve questions, although time and space efficient, does have one drawback; if a question needs to be enlarged in size, it is restricted to the original space allotted to it.

EXAMINATION PRODUCTION SYSTEM:

The "Examination Production" System is designed to allow the teacher to produce complete, line justified, examinations, ready to either give directly to the student or to mass duplicate. It allows the direct use of any and all questions included in the "Question Data-Base" as well as being able to add and mix new questions at any time. The output from the system can either be printed directly, or the system will generate a number of text files which are compatible with many commercial word processor systems such as Applewriter, Screenwriter, Executive Secretary, etc.

When the test is being composed, answer choices are randomly sorted. Running the system twice with the same questions, will produce two entirely different examination keys. In addition, the system automatically produces a printed examinatin key and projects an estimated difficulty factor for the examination based upon previous norm-reference information stored by the system. The teacher then has the ability to compare current class performance level with those of previous classes. In multi-class or multi-teacher situations the norm-referenced aspect of the system may allow for direct comparisons and possible teacher evaluation by comparing actual and expected grade or test averages.

Provision is made for selecting questions on the basis of any of the code information stored for each question, key terms, or difficulty level. Questions may also be chosen randomly. The random-choice works best in objective or

criterion based examinations where each question file covers a specific subject-concept area. If the question files are arranged according to concept areas, the norm-reference information on each question can generate a concept-area difficulty level .

The ability to prepare text files for use in the commonly available commercial word processing systems proves helpful in allowing for insertion of special printing characters and other manipulation of the text material. This includes the using of various spelling checkers to proof-read the test for errors.

TEST SCORING & ANALYSIS SYSTEM

This system consists of a series of programs which allows for the machine reading, scoring, and analysis of examinations containing multiple choice, True /False, and matching questions. This system is not able to grade essay or completion type questions.

The grading of examinations requires the use of a card reader or scanner. The most commonly available units which work with this system are made by Chatsworth Data, True Data, Mountain Hardware, HEI, Zweig, and Scantron Corporations. Input into the microcomputer is either through an Apple specific interface or through an RS232 Serial interface card. Student examination cards or forms are scanned by the reader and the student score recorded in a diskette file. If previous examinations have been taken by the student, the system displays what the student's current grade average is, and how far the current test score is above or below that average. In an attempt to improve feedback to the student, examination grading was attempted immediately as the student handed in the answer form. The student, upon completing the examination would feed his or her response card into the card reader and within 10 seconds would have an item listing of correct and incorrect responses and their score on the test. The general response of the student to this method was negative. Students felt more comfortable waiting until scores were posted later in the day or the next day. The immediate grading of the examination also tended to reduce reinforcement of learning since, during the normal waiting period, most students would spend time reviewing question answers in an attempt to determine, on their own, what questions were right or wrong.

Following the entry of all examination cards for a particular test, the student responses can be analyzed in the following fashion: (1) Question Item Analysis,(2) Norm-Reference Analysis, and (3) Subset Analysis.

An item analysis of each individual question is the customary method of evaluating test performance. The system performs an item analysis on each question and includes the typical percentage correct and percertages for each answer choice. It also includes a 2 point biserial correlation value for each question, the average test grade for those students picking each answer choice and a Student's "t" value for the difference between the test averages of those having the question correct versus those having it incorrect. When more than 50% of the class misses a question and where the "t" value is below 2, the question is flagged as possibly being defective. Using the percentage correct as indicative of difficulty level, the question data-base is automatically updated thus generating a self-developing norm-reference system. Using the CAS system over the past 4 years, Kuder-Richardson values above 0.8 have been consistently produced in over 70 different examinations.

The norm-reference analysis is a specialized type of grade analysis which is primarily student oriented. In this analysis the response for each individual student is compared to the norm-reference value on a question by question basis. The numbers and percentages of correct and incorrectly answered questions for norm-referenced and non-norm-referenced questions are compared. This information can be used to determine how dependent students are on old, circulating questions and how new questions compare, difficulty-wise, with old or previously used questions. The analysis also determines the difficulty levels for each question missed or answered correctly and gives an indication whether or not the student is missing consistently difficult questions or whether their performance is random, indicating a high dependence on guessing. Table 2 illustrate the difference between item analysis and norm-reference analysis. Using the norm-reference information, the course instructor is better equipped to counsel students on how their study habits need to be changed.

The first student in the norm-reference analysis would be considered a student with well rounded study havits since both old and new questions were answered about the same, and in fact, the new questions were of a more difficult nature as evidenced by their lower average reference values. The second student, however, although showing an average test performance in general, shows a high dependence on studying old test materials and avoiding new materials or studying from the textbook. In addition, the new questions

Table 2: Item Analysis and Norm-Reference Analysis

QUESTION 1 ANSWER IS (D) ** From Anatomical Terms - 34

ANSWER CHOICE	PERCENT	NUMBER	GRADE AVERAGE	STANDARD DEVIATION	POINT BISERIAL CORRELATION
A)	5 %	12	49.24	8.76	
B)	4 %	8	50.37	8.55	
C)	13 %	27	49.96	9.91	
D)	78 %	164	64.90	11.84	0.77 (Significant)
E)	0 %				
-)	0 %				

PERCENT CORRECT = 78% t-VALUE = 6.91

- - - -

as a "dropped" score for exclusion in calculating grade averages. Projections of future test scores needed to attain particular final grades can be made based upon remaining tests. Among the items of information stored for each student is a code field containing up to 5 different code levels. Each digit of the code field can be assigned to a particular area. Thus a teacher can have one grade file with all their students recorded in it, but identified by class period, etc through the use of the code field. In one application involving a large Anatomy and Physiology course, one field was used to identify the lecture section of the course, one to identify the laboratory section, another to identify the student's academic Division, another to identify academic major, and the last code field to record the student's combined SAT score. The Grade Analysis program utilizes these code fields and will print a comparison analysis of mean, standard deviation, maximum, and minimum scores by each code within each code field. Using this aspect of the program has allowed an assessment of the performance of the various laboratory instructors and the impact or relationship of SAT scores to grade performance. The coding by student major also proved valuable for the various program directors to evaluate how their students were doing in the course in comparison to other majors.

One difficulty in evaluating potential differences between student performance information from one year to the next is the fact that scores must be brought to a common base before valid comparisons can be made. The fact that one year the class average is 75 and the next year is 78 may reflect a difference between the student groups, or may only reflect a difference in the question difficulty from one year to the next. To avoid differences due directly to testing, the norm-reference information becomes vital. With the current system, if a test is given which, according to the norm-reference information should produce an average of 78, and the class averages only 72, it suggests some qualitative difference between the two populations. If an SAT score analysis and/or a student major grade analysis is also performed, it can be determined whether the differences are attributable to either of these two factors. The data might also show, however, that the assigned laboratory instructors, which can change from year to year, might have the greatest effect.

The system also includes a variety of grade curving routines which allow grades to be standardized from one year to the next or to a standardized performance level. Most instructors who choose to curve a test really only adjusts grades upwards through the expediency of dividing the raw scores by a lower point total. This procedure is biased according to the position of the student with the grade scale. The curving module of this system allows for grade adjustments according to three distinct models. In one model, the mean and standard deviation set the base grade scale and the mean plus or minus one standard deviation is the C range. The scores between one and two standard deviations represent the B and D range, and A and F are beyond the 2 standard deviation mark. In this model the user determines the final mean and standard deviation and each individual score is adjust to the save relative position in the scores as the initial score had. The second model uses a 90, 80, 70, 60 grade scale for the final grades and moves each score into the same relative position assuming the 70-80 range represents the C range. In the third model the user sets both the beginning and ending breakpoints for each grade level. Using any one of these models allows for a rapid manipulation of student scores to virtually any desired average. With the use of norm reference information and the projection of expected performance levels from previous use history, tests can be curved to equal previous or anticipated results. Doing this type of curving allows for an accurate determination of course difficulty from one year to the next.

Another factor which is vital to the exchange of information between teacher and student is the necessity to present to the student their grade information in such a form that it not only describes what their test scores are, but also where their weaknesses are and what they need to obtain a desired grade level. A Grade Report program is included in this system which has the ability to print test scores by the traditional percentage/point basis, but also allows for subsetting and for listing grades by Stanines. Using Stanine values eliminates some of the difficulties in judging the relative performance of a student within a group of students. The student codes included in the gradebook information may also be used to selectively produce customized grade reports for particular groups of students. In one application, one of the code levels was a code for student advisor. Reports were then prepared for each advisor on the progress of the students assigned to them. Reports with or without name information can be prepared to allow for the traditional "posting" of test scores. A summary list of grade distributions and test averages are also automatically included at the end of each grade report.

The reaction of the students to the information generated by the system has been mixed. They generally are in favor of the quick (1 day) return of test results but did not like the immediate reporting of test score[illegible]he division of tests by content or concept area initially was met with the comment: "Yes! but how did I do on the examination?" They wanted the security of that one unified grade. They did, however, find that the areas of weakness on examinations as shown by subsetting, were different from what they had assumed were their weak areas. A typical response was: "Well I thought I had done about xx% but I thought I had done better in this area then I had and did better in that area." In addition, there was a significant movement away from using old, circulating copies of examinations as comparisons of performance on each type of question was made available to the student through the norm-reference analysis. The analysis by academic major has prompted one department to revise their admissions criteria after repeated grade analysis reports showed that their students were performing significantly below all other groups of students in the same course and at the same time.

SUMMARY

The microcomputer and the use of CAS can be used very effectively to reduce the work load of a teacher by performing many of the mundane, tedious tasks of classroom administration. The system has also allowed an expansion in the types of grade analysis and has improved and made more effective the counseling of students. It has also allowed the development of automated test construction, and the utilization of fully norm-referenced test banking with a wide range of question types. These changes have substantially effected the methods used in the classroom to evaluate student performance.

The Capabilities of

the MicroTICCIT Authoring System

Jeffery J Howell

Microtel Learning Services

ABSTRACT

This presentation discusses the capabilities of the MicroTICCIT Authoring System as used in a telecommunications industry training facility. The MicroTICCIT Authoring System runs on a Data General Corp. microEclipse computer with network distribution to IBM PC microcomputers. Courseware developed for the British Columbia Telephone Company is cited for references and examples. Major topics include the use of MicroTICCIT in the telcommunications industry, system configuration, workstation equipment, training development software, instructional management system, data summary from course implementation, and conclusions for use in business and industry.

INTRODUCTION

Four years ago British Columbia Telephone Company (B.C. Tel) faced the problem of meeting the increased costs associated with training employees in an industry that uses rapidly changing technology. A needs analysis study indicated that Computer Based Instruction's (CBI) potential benefits outweighed the associated short-term start-up costs and could be an effective solution to their training problem.

B.C. Tel determined that Hazeltine Corporation's TICCIT System (Time-shared, Interactive, Computer-Controlled Information Television) best met the criteria for its CBI System. A test course decreased the total training time by 50% and increased learner-retention by 13%.

Other results of B.C. Tel's study were: CBI course development is not economically viable for every organization; cost-effective solutions can involve contracting externally for tailored or custom courses; off-the-shelf courses in some areas are available but quality varies widely; and contracting for space on the CBI-delivered courses of external suppliers is possible.

In January 1984, because of the success of implementaion at B.C. Tel, the CBT group of Microtel Learning Services was established. Microtel Learning Services now distributes courseware on the latest TICCIT CBI System, known as MicroTICCIT, to B.C. Tel and the telecommunications industry at large. The following describes the use of the MicroTICCIT system by Microtel Learning Services.

SYSTEM CONFIGURATION

MicroTICCIT is available in three standard hardware system configurations (System I, II and III) as well as custom-designed configurations for special applications.

System I configuration is designed for up to 10 users. It is suitable for delivery and editing of courseware and for small-scale development of new courseware. The elements of System I are detailed below:

* Data General MicroECLIPSE S/20 host processor with 256 Kbytes RAM
* 15 Mbytes of fixed disk storage for on-line system software, student records, and courseware
* 1.2 Mbyte floppy diskette for loading or dumping data from fixed disk
* Data General operator console CRT for performing system startup, shutdown and backup routines
* ARCNET Local Area Network communications interface which allows remote workstation locations
* A modem for interfacing to the maintenance service network

System II configuration is designed for medium-scale CBI efforts. This system supports approximately 40 users for authoring, learning or management activities. The elements of System II are detailed below:

* Data General MicroECLIPSE S/20 host processor with 256 Kbytes RAM
* 50 Mbytes of fixed disk storage for on-line system software, student records and courseware
* 15.4 Mbyte cartridge tape drive for loading or dumping data from fixed disk
* Data General operator console CRT for performing system startup, shutdown and backup routines
* ARCNET Local Area Network communications interface which allows remote workstation locations
* A modem for interfacing to the maintenance service

System III configuration is a family of systems designed for full optimization of the MicroTICCIT Authoring Software in large-scale CBI efforts. System III supports up to 64 workstations for either student, instructor, or author functions. The elements of System III are detailed below:

* Data General microECLIPSE S/20 host processor with 256 Kbytes RAM
* 15 Mbytes of fixed disk storage for rapid "swapping" of data from disk to processor
* Additional fixed disks for on-line storage of software, student records and courseware
  - System III-A: 50 Mbytes
  - System III-B: 73 Mbytes
  - System III-C: 147 Mbytes
* Tape drive for loading or dumping data from the fixed disks
  - System III-A: 15.4 Mbytes Cartridge Tape
  - System III-B: 25 Mbytes Streaming Tape
  - System III-C: 25 Mbytes Streaming Tape
* Data General operator console CRT for performing system startup, shutdown and backup routines
* ARCNET Local Area Network communications interface which allows remote workstation locations
* A modem for interfacing to the maintenance service

WORKSTATION EQUIPMENT

The standard student/author workstation consists of an IBM Personal Computer equipped with a Sony Profeel full-color display, an expanded key board and a host of other optional interactive devices.

The IBM PC comes with 64 Kbytes of RAM. The display controller and communications interface for the PC micros with the Eclipse mini require two memory expansion slots. That leaves three slots available for expanding the PC for stand-alone use.

The picture element (pixel) resolution for the Sony display is 430H x 204V. The screen refreshes at sixty frames per second to provide a flicker-free, stable display. The Sony Profeel proves to be excellent display for text, graphics and videodisc or tape. It allows long-term viewing or authoring with little fatigue or eye strain.

MicroTICCIT supports a sophisticated graphics capability, using a command-based editor for simple shapes and a pixel-level editor for shading and details. A graphics digitizer is available to increase production rates. Several thousand colors can be obtained by manipulation of hue, intensity and saturation of the display image.

The keyboard is detached and has an expanded keyset. The keyset includes the standard items as well as an editing and cursor control keypad, dedicated function keys for controlling the authoring software and eleven user definable function keys.

A videodisc or videotape player can be added to the system to allow motion or still-frame video and computer-generated text and graphics on the same screen, either simultaneously or separately. There is also an audio option available for recording, editing and converting speech into compressed digital formats.

TRAINING DEVELOPMENT SOFTWARE

The heart of MicroTICCIT is the ADAPT Authoring System (the name comes from the philosophy of "adapting" to the authors skill level). The ADAPT Authoring System uses system level prompts and built-in macro-functions for course development. There are three levels of ADAPT authoring available (the first two levels are under development at the time of this writing). The first level makes heavy use of menus, prompts and fill-in-the-blanks for system automatic screen display construction, response analysis, symbol and branch command construction. The second level makes less general use of the system automatic functions but, retains many of the menus and prompts. The third level retains some prompts but, relies on the developer for most of the screen presentation construction. The highest level of ADAPT allows more freedom and ease of course construction for the experienced developer. While the lower levels provide more speed and efficiency for one less experienced. ADAPT provides built-in instruction and on-line helps for all authoring functions.

The system uses a built in Rule, Example, Practice (REP) Instructional Systems Development (ISD) model similar to the U.S. Navy's Instructional Quality Inventory (IQI) Model. The built-in REP Model can be used as a system level automatic function to give inexperienced authors a more rigid and instructionally sound framework for course material development. The REP Model can be overriden by more experienced authors.

Authors create text displays exactly as they will be seen by the student, with the desired colors and backgrounds incorporated directly into the display as it is typed. A set of "word processing" features are available to speed text entry.

ADAPT supports graphic production with the following techniques: direct entry of a hard-copy line drawing by video digitizer camera; typing of symbols from character "dictionaries" supplied with the system or defined by authors for special applications; use of commands or the light pen to "draw" lines, circles, and other standard figures; and use of mathematical expressions to plot curves and functions. A graphics animation package is currently under development.

INSTRUCTIONAL MANAGEMENT SYSTEM

In addition to the instructional development and delivery capabilities, the MicroTICCIT System provides an integrated Instructional Management System. Data is automatically collected and made available to instructors, courseware developers and training administrators. Varying types and quanities of data may be collected, depending on the particular MicroTICCIT system configuration and user requirements.

The performance data can be used to produce a number of standard reports as part of the MicroTICCIT Instructional Management System. The reporting programs build and use a variety of summary files which are then used to display or print desired reports. The summary files themselves constitute an additional database which may be used for subsequent applications.

DATA/IMPLEMENTATION SUMMARY

At the time of this writing, Microtel Learning Services has the results of the first two courses implemented by B.C. Tel's Education Center. Both courses are designed for technical and maintenance personnel who will be servicing and repairing digital equipment at the card replacement level and who require an understanding of the fundamental concepts underlaying digital systems and computers.

The first course is Digital Logic. The study consists of a control group taking the course in a self-paced print format and an on-line self-paced CBI version. There were 113 students who took the print version of the course and 63 students who took the CBI version. The control group took a mean time of 8.5 days to complete the course with a mean final grade of 76%. The CBI group took a mean of 4.4 days to

complete the course with a mean final grade of 93% (see figure 1).

Completion Time Summary

	TICCIT (N=63) DAYS	Paper Version (N=113) DAYS
Mean	4.4	8.5
SD	0.7	1.8
Fastest	3.0	3.0
Slowest	5.0	13.0

Student Performance Summary
(Average of first attempt test scores)

	TICCIT (N=63)		Paper Version (N=113)	
	Units	Final	Units	Final
Mean	95	93	85	76
SD	5	8	5	12
High	100	100	100	100
Low	81	58	73	33

Figure 1. Digital Logic Course Comparison

The second course is Computer Fundamentals. The pilot version of this course was tested on March 11 and 12, 1984. There were 28 students, all male, with an average age of 29 years; the group averaged 7 years employment with the company and 3 years in their current position. The majority of the students had little or no experience with MicroTICCIT, CBI, digital logic or computers of any kind.

The 28 students received an overview of the CBI system and proceeded with the course. The system recorded the student time on-line on each unit, segment and section of the course. Time away from the terminal was not recorded as students are required to "log-off" then "log-on" when going on break. The time spent on-line ranged from a low of 2 hours, 56 minutes to a high of 10 hours, 35 minutes. The average time to complete the course was 6 hours, 22 minutes (see figure 2).

Previous Experience

Yes	No	Some
6 Students	16 Students	6 Students
2 With TICCIT	57% of Population	2 With Computers
2 With a Digital Logic Course		4 With Previous Courses
2 With O.J.T.		

Present Position

Central Office Maintenance	4
Instructor	1
Sales	1
Network	1
Engineer	1
Craftsman	2
Installer	6
Canadian Telephone & Supply Equipment Installation	4
	8

Educational Background

Secondary	25
University	2
Technical	9
Community College	3
Correspondence	1
Evening Classes	3

Figure 2. Computer Fundamentals Course Target Population

Microtel Learning Services is currently using the MicroTICCIT System to develop two more courses: "Managerial and Engineering Economy" and "Fundamentals of Basic Transmission". Several more courses are in the proposed development stage.

CONCLUSIONS

Several conclusions can be drawn from using and implementing courses developed on the MicroTICCIT Authoring System.

First, contrary to advertisements, there is no easy authoring system. Authoring CBI courses, whether for micros, minis or mainframes, is a complex process which requires an understanding of human learning and an attention to detail not found in any other instructional medium. "Easy inputting" -maybe; "easy authoring" -definitely not! CBI requires experienced developers, designers and support personnel to produce cost-effective solutions.

Hardware costs are coming down. B.C. Tel has experienced a 50% decline in hardware costs over the last 2 years coupled with an increase in performance. On the other hand, course development costs, mostly because of salaries, are rising. The development of techniques to reduce development time and costs will provide a new challenge to us in the future.

Of great importance and a reinforcement of what others have stated in the past, the learning environment has a measurable impact on the effectiveness of training delivery. Human factors design considerations must be employed in the CBI classroom for a cost-effective implementation of courseware. A subject-matter expert, acting as a course manager, is needed to greet, register and provide the human touch students need. Finally, CBI course development is not economically viable for every organization. However, that does not mean CBI is not possible. Cost-effective solutions may involve contracting externally for custom courses. Contracting for seats on the CBI-delivered courses of external suppliers is another way. Off-the-shelf courses in some areas are available.

Industrial commitment to CBI continues to grow. We at Microtel Learning Services will continue to use and experiment with the MicroTICCIT Authoring System in an effort to meet the telecommunications industry's commitment. Used with skill and creativity in applications suited to its strengths, the medium combines the strengths and patience of great teachers. The key to its effectiveness is to find the right applications and to efficiently integrate it within the mainstream of industrial training.

TOWARDS THE DEVELOPMENT OF AN IBM PC AUTHORING SYSTEM

George Reed
Louisa Frank
Deborah Richards
Nancy Balogh
University of Delaware/Office of Computer-Based Instruction

ABSTRACT

There are enormous committments of both time and programming resources involved with the development of computer-assisted instructional (CAI) software, especially good quality educational programs.

A well-implemented authoring system is one way to decrease the time it takes to create a complete computer-based lesson. The CAI author can produce the software, responding to prompts for input from the system, without the aid of a programmer, while able to make full use of the capabilities of the computer.

This paper will review the considerations that one university has outlined as essential when developing a complete authoring system for CAI authors.

INTRODUCTION

Education materials for the personal computer have not yet gained the widespread use they deserve. One reason for this is the difficulty of producing the programs. Estimates for the time it takes to develop one hour of computer-assisted instructional (CAI) materials range from 100 hours to upwards of 800 hours. Such factors as the complexity of the instruction, the interaction, and the extent to which the capabilities of the computer are employed must be figured into the development time. High quality CAI materials, ones that take full advantage of machine and involve discovery and simulation, cluster at the high end of the development timeline. In fact, one fear for developers of innovative CAI is that the targeted hardware will no longer be popular by the time the product is out on the market.

One way to decrease the time it takes to create a complete CAI lesson is to use an authoring system. Unlike an authoring language, which still requires a user to learn programming logic and constructs, an authoring system prompts the user for input and choices. All the programming structure is built into the authoring system but hidden from the user. Such a system allows a nonprogrammer to create a complete, working lesson.

AUTHORING SYSTEMS

Developers at the University of Delaware gave careful thought to the qualities desired in an authoring system. The following list of criteria summarizes some of the major considerations in choosing an authoring system.

1. It should be easy to use for the nonprogrammer. Either menu driven or prompt driven systems enable the novice user to learn the system and produce software quickly.

2. It should not be limiting. There should be faster ways of communicating with the editor than through menus, so that the experienced user is not frustrated with a slow paced interaction.

3. The authoring system should not limit the types of instructional strategies to pre-determined lesson formats. The system must be flexible and allow the author to create any kind of instruction.

4. Ideally, an authoring system should be developed around a popular programming language. Then a user with programming experience would be able to use the authoring system for the more mundane programming tasks. If more intricate structures are deemed necessary, the user could leave the authoring system and go directly to the programming language.

5. An authoring system should provide templates for most of the repetitive programming needs of a lesson, such as a menu, a multiple choice question driver, sequencing of the lesson's displays, and branching based on users' responses.

6. A good system must allow creation of screen displays, with full graphics and text options available to the author. It should be easy to revise previously created displays and to connect them to form a complete lesson.

7. An authoring system should take full advantage of the capabilities of the system on which it operates, for example, graphics resolution, available color, and fast mathematical calculations.

For creating lessons using the videodisc, some added features are needed in the authoring system.

8. There must be a method which allows easy access to the videodisc content, such as a database (video index) from which frames or sequences can be called by name.

9. The authoring system must allow the creation of text and graphics frames directly on the video image.

CREATING AN AUTHORING SYSTEM

The IBM project at the University of Delaware reviewed several authoring systems but did not find one that satisfied all these criteria. Therefore a decision was made to create an authoring system for use on all university IBM PC software development projects.

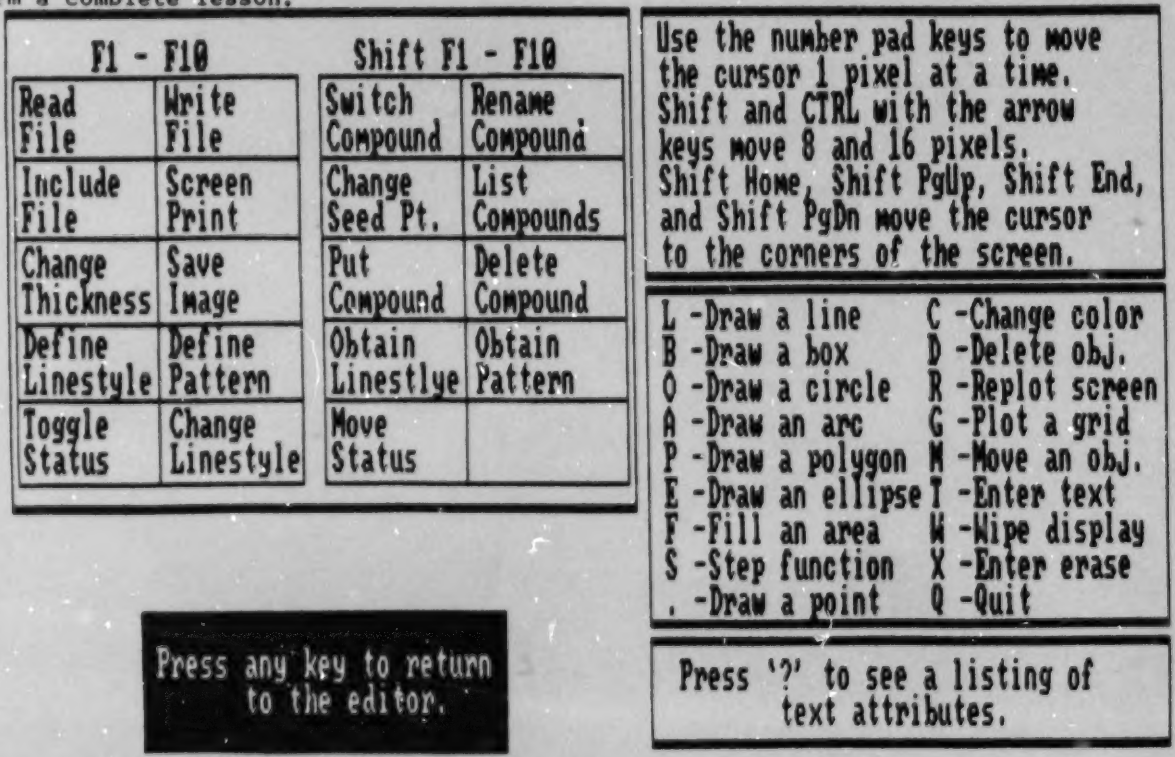

F1 - F10
Shift F1 - F10
Read File
Write File
Switch Compound
Rename Compound
Include File
Screen Print
Change Seed Pt.
List Compounds
Change Thickness
Save Image
Put Compound
Delete Compound
Define Linestyle
Define Pattern
Obtain Linestlye
Obtain Pattern
Toggle Status
Change Linestyle
Move Status
Use the number pad keys to move the cursor 1 pixel at a time. Shift and CTRL with the arrow keys move 8 and 16 pixels. Shift Home, Shift PgUp, Shift End, and Shift PgDn move the cursor to the corners of the screen.
L -Draw a line
B -Draw a box
O -Draw a circle
A -Draw an arc
P -Draw a polygon
E -Draw an ellipse
F -Fill an area
S -Step function
. -Draw a point
C -Change color
D -Delete obj.
R -Replot screen
G -Plot a grid
M -Move an obj.
T -Enter text
W -Wipe display
X -Enter erase
Q -Quit
Press any key to return to the editor.
Press '?' to see a listing of text attributes.

FIGURE 1. Main HELP Display

Initial efforts began with a screen display editor(the graphics editor) which at first allowed only text and simple graphics entry. It has since been expanded to include full scale color graphics. The overall objective is for the user to be able to make all major editing changes to the lesson while viewing the displays on the screen. The graphics editor became the kernal of the authoring system. As new features were developed, they were incorporated as features of the screen editor. The screen editor now exists in two distinct versions, known as the Graphics Editor (GE) and as the Monochrome Graphics Editor.

The Graphics Editor

The Graphics Editor is a prompt-driven utility for creating a screen containing text and graphics. The screens generated by the user are stored as Pascal procedures. These procedures may be called up again by the editor and modified at any time. Because they are stored as editable code, these displays may be altered by anyone familiar with Pascal. Thus features not currently supported by the authoring system can be added easily without losing the advantage of the editor. Features that one might add include scoring variables, mid-screen pauses, timing, and complex animations.

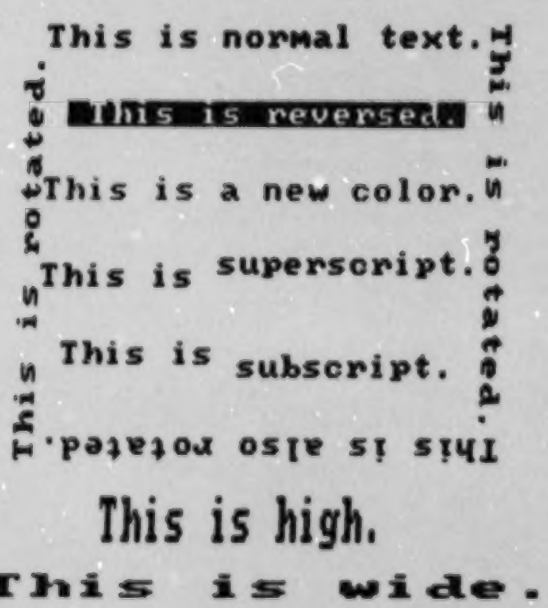


FIGURE 2. Text Formats

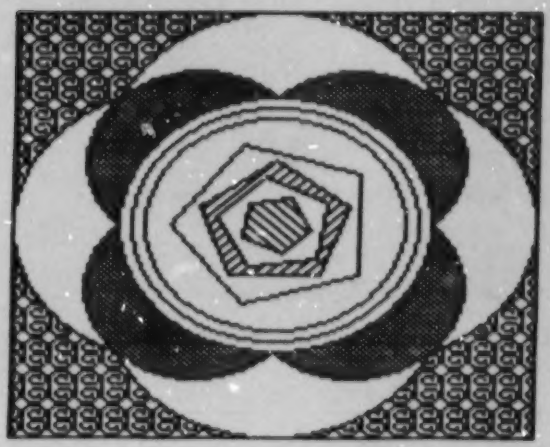


FIGURE 3. Sample Graphics

Figure 1 is a help display that is available from the main graphics editor page. This display shows the features that are now supported in the editor. A more detailed explanation of some of the features follows.

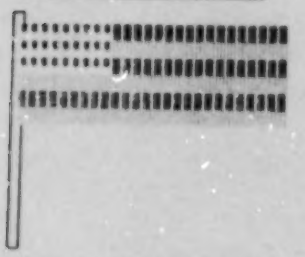
Text

Text entered on the screen may have several attributes. For example, it may be rotated or reversed, of variable height and width, in various colors, and may contain sub and superscripts. The text and its attributes can be edited at any time by calling the frame to the screen again. Figure 2 shows some of the available text formats.

Graphics

Available graphics elements include lines, boxes, circles, arcs, ellipses, and polygons. Lines may be defined as having any thickness, and can be assigned a user defined pattern. Also, any closed figure on the screen may be filled with a user-defined pattern. Figure 3 shows a sample of the graphics possible.

GOVERNMENT DOCUMENTS



Copyright 1984 by the University of Delaware

All rights reserved.

Press RETURN to continue.

FIGURE 4. IBM Monochrome Screen created with the Graphics Editor

Objects

The Graphics Editor also allows the user to create compound objects. These compound objects may consist of any combination of text and graphics. Once created, the compound object is treated as a simple object by the editor. Both simple and compound objects may be moved or deleted by the user. In addition, the compound object may be stored in a library for use in other displays. Also an entire screen can be called in as part of a new display and edited.

Screen Modes

The graphics editor can be used in either low resolution mode or high resolution. Low resolution supports a screen size of 40 characters and 24 lines and also includes color options. The high resolution mode supports a screen of 80 characters and 24 lines and is only available in black and white mode. Support for higher resolution/color modes is now being developed.

The Monochrome Graphics Editor

The monochrome graphics editor is designed for use with the IBM monochrome display adaptor. Like the color graphics version it allow the creation of text and graphics displays. However, the graphics capabilities are limited to special IBM characters. Graphics capabilities include boxes, lines, and fills. All graphics and text must be inserted on character and line boundaries, whereas the graphics objects in the full graphics version are positionable at any pixel location on the screen. For text entry there are several attributes, such as reverse video text, underlined text, bright text, and underlined-bright text, however, the text cannot be sized or rotated. The monochrome graphics editor supports a screen size of 80 characters by 24 lines. Figure 4 shows a screen display produced on the IBM monochrome screen using this graphics editor.

Sequencing

The sequencing of the lesson is currently accomplished using a templated Pascal module containing case statements which direct the flow of the lesson. One such module is shown in figure 5. A programmer with only limited knowledge of Pascal is able to edit the case statements in this module and create the sequence for a lesson.

```
MODULE  DO_PAGE_MOD;

( $INCLUDE: 'sequence.int')     (* sequencing variables *)
( $INCLUDE: 'grinter.pas')      (* UD graphics routines *)

(*-------------------------------------------------------*)
(* Include here any pascal files with units to be       *)
(* called by the DO_SEQ1 procedure                       *)
(*-------------------------------------------------------*)

PROCEDURE PITITLE; EXTERN;
PROCEDURE PIMENU; EXTERN;
PROCEDURE PI1PO1; EXTERN;
PROCEDURE PI1PO2; EXTERN;
PROCEDURE PI2PO1; EXTERN;
PROCEDURE PI2PO2; EXTERN;
PROCEDURE DO_SEQ1 (VAR CODE : INTEGER);

(*-------------------------------------------------------*)
(* When DO_PAGE is entered, var. MAINU = section to be   *)
(* called.  Branches are set.  Control transferred to    *)
(* section proc.  Upon exiting section proc., ENDPAUSE is*)
(* called to determine next section.  DO_PAGE is called  *)
(* repeatedly by driver proc.MAIN until next section=QUIT*)
(*-------------------------------------------------------*)

  BEGIN
    INITIALIZE;  (* set up initial conditions *)
    WHILE MAINU <> QUIT DO
    BEGIN
      GET_NEW_DISPLAY  :=  TRUE;
      CASE MAINU OF
           TITLE    : PITITLE;   (* file pititle.pas *)
           MENU     : PIMENU;    (* file pimenu.pas  *)
           SEC1PO1  : PI1PO1;    (* file pi1po1.pas  *)
           SEC1PO2  : PI1PO2;    (* file pi1po2.pas  *)
           SEC2PO1  : PI2PO1;    (* file pi2po1.pas  *)
           SEC2PO2  : PI2PO2;    (* file pi2po2.pas  *)
           OTHERWISE
               WRITE('MAINU = ',MAINU);
      END;  (* do_page case *)
      IF  GET_NEW_DISPLAY  THEN
          ENDPAUSE (DISPLAY_KEYSET);
    END;  (* while *)
    CODE  := 1;  (* okay *)
  END;   (* procedure do_page *)
  (*-------------------------------------------------------*)
  END.   (* module domain *)
```

FIGURE 5. Sample Code generated by Graphics Editor

The structure of the program allows linear forward and backwards movement in the lesson as well as branching based on special key presses requesting help or review or based on a student's response to a question. The more experienced programmer can modify this module to allow for other types of branching, for example, one based on timing, on a student's score for a series of questions, or on any other criteria. Work is now underway to provide an automatic sequencing facility so that users with no knowledge of programming will be able to create a complete lesson.

Judging

Routines written for the authoring system include judging mathematical expression, short answer inputs, and screen location. The math answer judging will accept student answers within a preset tolerance. It will also accept simple math operators and evaluate the student's answer, printing the numeric value on the screen before judging whether it is correct. The short answer judging looks for an alpha numeric string of a preset length. It is currently used for short word answers and for single keypress judging from menus and multiple choice questions. The screen location judging treats defined areas of the screen as the input of multiple choice questions.

Future work on judging will include expansion of the math judging to allow more complex operations, such as powers through superscripting and trig functions. Spelling checking on word answers and sentence and keyword judging are also planned. The screen location judging will be able to return single pixel locations for use in graphing.

Additional Features

The current authoring environment is rich in routines that allow the user to be creative while designing a good instructional lesson.

An editor for fill patterns allows a user to create a pattern of pixels, from a size of 1 by 1 up to 25 by 25. Each pixel may be of any allowable color. This pattern is then assigned an index number and can be used to fill any closed geometric shape on the screen. The pattern in figure 6 was created in this manner.

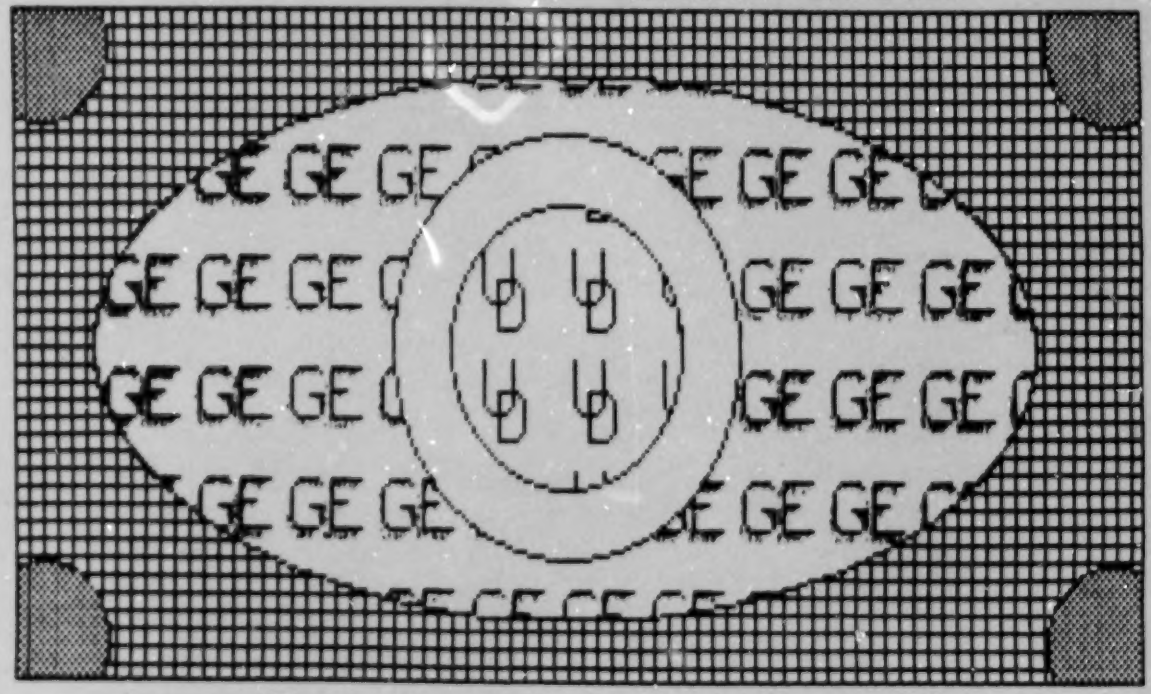


FIGURE 6. Examples of Filled Shapes

A tray file can display a series of created screens for proofing and review before compiling. Thus the displays of a lesson can be created and reviewed without compiling the lesson. This is a tremendous time saver during development.

A character set editor lets the users create their own 8 by 8 characters. Several characters can be created on one editor display so that pictures larger than 8 by 8 can be made. This facility is especially useful in creating the special characters needed for mathematics, foreign language, or engineering lessons.

CONCLUSION

The future of CAI development depends on the development of good authoring systems, to reduce both the time and the cost of production. The University of Delaware's Authoring System is one attempt to provide a complete development environment for CAI authors.

Analog to Digital Conversion

Robert S. Tannenbaum

Hunter College of the City University of New York

David C. Tannenbaum

Rensselaer Polytechnic Institute, Troy, New York

ABSTRACT

We have designed an analog data acquisition and conversion circuit which is relatively simple to build and interface to any microcomputer. We first explain the difference between digital and analog signals and why and how we convert from one to the other. Then we describe some fundamental software for performing conversions. Next we explain some basic hardware considerations such as range, calibration, resolution, speed, and multiplexing. Finally, we describe the circuit we have designed and some software to control it.

INTRODUCTION

Computers are becoming more and more common. Today they can be found in large numbers in many schools, colleges, and other training centers. Unfortunately, many of these computers are not being used to their fullest. Computers are effective number crunchers, word processors, and teachers, but they can do so much more. For example, computers can run entire experiments. They need not merely process the data from the experiment. They can also operate equipment, monitor the environment, and coordinate complex complete operations. In Computer Assisted Instruction, computers need not interact with the learner only through the keyboard and the monitor. They can control a wide range of peripheral devices for demonstrations (see Tannenbaum and Tannenbaum (1983)). They can also acquire data from a wide range of input devices which permit students to respond in appropriate and realistic fashion while they learn. This paper focuses on computer acquisition of analog data.

We have designed an analog data acquisition and conversion circuit which is relatively simple to build and interface to any microcomputer. We will first explain the difference between digital and analog signals and why and how we convert from one to the other. Then we will describe some fundamental software for performing conversions. Next we will explain some basic hardware considerations such as range, calibration, resolution, speed, and multiplexing. Finally, we will describe the circuit we have designed and some software to control it.

ANALOG DATA

Data is usually classified as either analog or digital. An example should clarify the difference between analog and digital.

> Imagine a simple home circuit with a light bulb and a switch. The light bulb has two possible states. It can be on or off, but not both. This is a digital system because there are only discrete states allowed. And it is binary because there are only two states.
>
> If you replace the switch with a light dimmer, you have created an analog system. The light can now have an infinite number of states. There are no discrete states in an analog system. The system is continuous.

Almost all computers used in computer assisted instruction are digital. The data which they process and which they input and output must be represented in discrete, binary codes. But the real world is almost entirely analog.

Analog to Digital Conversion

We often convert analog data to discrete or digital form and back again almost automatically. For example, you can translate the infinite variety of spoken English into various combinations of just twenty-six alphabetic symbols. That is an analog-to-digital conversion. And you can read this passage aloud. That is the reverse process, a digital-to-analog conversion.

Some sophisticated systems (such as your brain) can handle both digital and analog signals. Many systems (such as most computers) can handle only one or the other type of signal, but not both. For that reason, we need a circuit and some software to control it which will gather analog data from the real world and convert it to digital form for processing by your computer.

By using transducers, many kinds of physical properties can be represented by voltages and thereby used by the computer. For example, the diaphragm in a barometer, instead of moving a needle, could move a potentiometer which produces a voltage proportional to the pressure.

SOFTWARE

Let's look at some software used with analog to digital converters. In many applications, routines written in a higher level language are sufficient. If speed is not a factor, you can use an interpreted language. If that is too slow, you may compile your program. In some applications you may need to use an assembly language. We will use BASIC in our examples because it is so widely known and available. If you prefer, you can easily convert our programs to any other language or dialect.

Peek and Poke

The two elementary functions you will need are PEEK and POKE.

PEEK enables you to move data from any location in the computer to the location associated with a variable named by you in a higher level language. PEEK is used to allow your program to make decisions based on the contents of a memory location not directly accessible with the high level language.

For example:

10 V = PEEK(1234)

results in whatever is currently stored at location 1234 being moved to (i.e., replaced at) the memory location associated with the variable V.

PEEK is a move from an unnamed computer location to one named by a higher level language program.

The reverse is true for POKE.

POKE enables you to move data from a memory location associated with a variable named by you in a higher level language to any permitted location in the computer.

For example:

20 POKE(5678,D)

results in the transfer of whatever is currently stored at the memory location associated with the variable D to location 5678.

Warning. POKE is potentially a very dangerous instruction. Because you can POKE to any location in the computer, it is easy to destroy essential system information and cause the computer to crash. You should therefore exercise care when using the POKE command.

Various dialects of BASIC implement the PEEK and POKE commands differently. If you are interested in using the hardware described in this paper, you will need to experiment with your language's PEEK and POKE.

With these two commands (peek and poke), you can cause the computer to send control codes to the analog-to-digital converter (A/D) and read the converted values. Here is a sample program:

```
10 REM This program uses PEEK and POKE with an A/D
   (
   (Bulk of your program and subroutines)
   (
990 REM Assume A/D is located at addresses 500 and 501
999 REM Subroutine to read the A/D follows
1000 POKE(500,1) : REM Start conversion
1010 FOR I=1 TO 100 : NEXT I : REM Delay
1020 POKE(500,0) : REM End conversion
1030 V=PEEK(501) : REM Read converted value and store
1040 RETURN : REM End of subroutine to read A/D
32767 END
```

This is, of course, a general example. The subroutine is designed to read an A/D converter with up to 8 bits of resolution. Address 500 is the control byte for this A/D. The control byte starts and stops the

conversion process. Address 501 is the data byte.

In your program, whenever you want to read the A/DC, you simply include a 'GOSUB 1000' in your code. The variable 'V' will contain the converted value when control is returned to the main program. The main program must now assign a meaning to this value. Does the number represent a temperature, a current, or maybe a light intensity?

Calibration

Your program must know how a change in the value converted relates to a change in the quantity of interest. For example, if you have a barometer connected to a potentiometer which regulates a voltage proportional to the atmospheric pressure, what is the relationship between a change of size delta V in the voltage and the actual change, delta P, in the pressure.

In mathematical terms:

What is $\Delta V / \Delta P$ or dV/dP
(V = voltage, P = pressure)

The ideal case is usually a one to one relationship such as $dV/dP = 1$ because it reduces the number of calculations. Let us assume, for the sake of simplicity, that you are using an 8-bit A/D. That is, the A/D hardware converts the analog signal into an 8-bit binary number. This means that you will always read a number between 0 and 255 (00000000 and 11111111) inclusive. Now let us further assume that you are interested in measuring pressures from 29.5"-Hg to 30.5"-Hg, standard atmospheric pressure. You therefore have a converted value of 0 represent 29.5" and a value of 255 represent 30.5". Now, your software needs an algorithm to generate a meaningful interpretation of all numbers between 0 and 255. A '213' read from the converter has little meaning as a barometric pressure. Therefore, you could perform the following calculation:

```
10 A = 29.5 + V / 255
REM A=Adjusted value, V=Value read from the A/D
```

the variable 'A' now contains the actual barometric pressure given in inches. This equation contains an offset and a scale factor. The offset is the starting point (29.5 in this case). The scale factor converts from one set of units to the other (from units of analog voltage to inches of mercury in this case). Here the scale factor is one; it appears implicitly in the equation. Scale factors not equal to one would appear explicitly as a coefficient of 'V.' You can now rewrite the subroutine that reads the A/D to do the mathematical adjustment and display the pressure.

```
999 REM Subroutine to read the A/D follows
1000 POKE(500,1) : REM Start conversion
1010 FOR I=1 TO 100 : NEXT I
1011 REM Delay until conversion is complete
1020 POKE(500,0) : REM End conversion
1030 V=PEEK(501)
1031 REM Read converted value and store in variable 'V'.
1040 A=29.5+V/255 : REM Adjust to give pressure in inches.
1050 PRINT"The current pressure is ";A;" inches of mercury."
1060 RETURN : REM End of A/D routine
32767 END
```

This more elaborate subroutine not only reads the A/D but also returns an adjusted value and tells the user the current pressure. You have included the scale and offset.

Input Range vs Resolution

Two other variables in A/D conversion are range and resolution.

Range — The number of units over which the analog variable can be converted to digital form.

Resolution — The number of digital units per analog unit after conversion.

The number of digital units which your A/D can produce is controlled by its number of bits. For example, an 8-bit A/D can only produce a maximum of 256 different values (0-255). For a given number of digital units you can choose to spread them over a wide range, with the result that you will have rather low resolution. Or you can choose to convert over a very narrow range and achieve much higher resolution. The barometric pressure conversion above has a range of 1 inch of mercury and a resolution of 1/256 of an inch (because we assumed 8 bits of conversion over the 1 inch range). If you decreased the range to 0.5 inches you would have twice the resolution over half the range. There is always a trade-off between range and resolution. The ratio of ΔV to ΔP (dV/dP) defines this relationship. See Figure 1.

It is possible to have nontrivial relationships such as $dV/dP = mX + b$ where 'mX + b' is the general equation of a line.

(Notice that if a=0, then the relationship is simplified and is the same as in the example above.) More complex relationships also exist.

HARDWARE

Now that we have examined some of the software used with A/Ds, let's turn our attention to the hardware aspects. First we describe some of the basic hardware concepts used in our A/D circuit. Then we present the circuit and describe in greater detail how it works.

A/D Conversion Methods

We have already indicated that the resolution of an A/D converter depends upon its input range. Resolution is also related to the method and rate of conversion. In all methods of conversion, the higher the resolution, the slower the rate of conversion. Just as in any other process, accuracy takes time. There are three widely used methods of analog to digital conversion:

- Successive approximation -- repeatedly compare the signal voltage with a known reference voltage. The known voltage is initially set at the center of the input range. If the signal is greater than the reference voltage, the high order bit is set to one. The input is now compared with a reference voltage midway between the maximum voltage and the center voltage. The same comparison is made and the second bit is set accordingly. The process is continued until all of the bits of the digital conversion have been processed. The technique is similar to a binary search algorithm. Doubling the resolution increases the conversion time only by the time of one additional comparison. When compared with the ramp comparison method (described next), the time savings are substantial. See Figure 2.

- Ramp comparison -- The input signal is continually compared with a reference signal which varies between the minimum and maximum of the input range. When the two voltages are equal, then the reference voltage represents the input signal. See Figure 3. This method requires many more cycles than the successive approximation technique. On the average, it takes half of the resolution size number of cycles to convert the input signal into its digital representation. Doubling the resolution doubles the average conversion time. Because of this, successive approximation is used much more frequently than ramp comparison.

- Dual slope integrating -- This is the most accurate of the three methods. A capacitor is charged by the input signal over a fixed length of time. It is then discharged at a known rate. The time it takes for the capacitor to be fully discharged is proportional to the input voltage and an integer multiple of the time constitutes the converted value. See Figure 4.

Standard 8 bit successive approximation A/Ds average a million conversions per second. Special flash decoding A/Ds give many millions of conversions per second. A dual slope integrating A/D gives high accuracy at the expense of conversion time. Typical conversion times for A/Ds of this type are in the hundreds of milliseconds or tenths of a second (about five conversions per second). We use an Intersil ICL7135 dual slope integrating A/D in our circuit for the the increased accuracy. The circuit could easily be modified to employ a different converter if greater speed were required.

Data Transfer

Not all A/Ds operate only with positive voltages. The A/D described above (under Calibration) uses only positive voltages and has an input range of 0 to 10 volts. 2.56V, 5.00V, 10.00V are each common full scale ranges for 8-bit A/Ds, but other values, including negative ones, are possible. These converters send the computer a single byte of data at a time. There are converters with more bits of accuracy, with sign bits, and with under-range and over-range signals.

The A/D can present data to the microprocessor in several ways. One of the most common methods is multiplexing.

Multiplexing

Multiplexing is transmission of each decimal digit of the converted value separately from the A/D to the computer. If you have an A/D which can convert to from 0000 to 1999 and it has multiplexed output, it will send the value of each decimal digit sequentially. To transmit 1935, the A/D would send a 1, then a 9, then a 3, and finally a 5. This is called a three and a half digit A/D because it has three full decimal digits, 0-9, and one digit with only two values,

0-1. The digits are usually in binary coded decimal (BCD) format. That is, each full digit is coded in binary from 0000-1001 and the half digit is coded 0 or 1.

The digits' values are sent sequentially, starting with digit zero (D_0 is the least significant digit). When the digit select line for a particular digit goes to the logical ON state, the data being transmitted is for that particular digit. A strobe line is used to eliminate errors due to timing problems which occur when several bits change their values simultaneously. For example, the digit select bit and the data bits for a particular digit should, theoretically, change from digit zero to to digit one instantaneously and simultaneously. But electronically they may be just slightly out of phase and cause "race condition errors."

Figure 5 shows a circuit with glitches (timing errors) due to race conditions. The glitches are the sharp spikes on the output waveform. Every glitch can cause an error in processing. This problem can be eliminated if the circuit can be told to 'ignore' the inputs while the race condition exists. The the strobe line is used to make the circuit wait until the race condition no longer exists.

To show how the strobe line works, look at the portion of our circuit shown in Figure 6. This circuit saves the value of the first four digits of the A/D converter in a 74LS670 16 bit RAM and displays the converted value on four 7-segment LEDs. The RAM is configured as four 4-bit words. (Note that a word need not be two bytes or 16 bits. A word is simply a logical grouping of bits.) The two OR-gates convert the four digital values into a binary code. The resulting binary code indicates which word is to be written to. The waveforms in Figure 7 show exactly how the strobe controls the timing. This figure is an expanded section of the waveform shown in Figure 6. You can see that the transition from one state to the other is not instantaneous. The transition time is a function of the particular components used and the environmental conditions.

If the RAM were being written to constantly, errors due to glitches would result. If the data for digit zero changed to the data for digit one an instant before the address changed, digit one's data would be stored at digit zero's address, or, even worse, total garbage would be stored. This is precluded by having the RAM only store data while the strobe is low. Because the strobe is only low (remember it is an active low signal) between the transitions, the race conditions are avoided. The 'write enable' input on the 74LS670 is the equivalent of a strobe input. The RAM only writes while it is "write enabled", i.e., when the strobe line is activated, it indicates that the address and data are valid.

A feature of this ram chip, the 74LS670, is that it can be written to and read from at the same time. You can be writing to one location and reading from another simultaneously. This is very useful in our application. We have the A/D constantly writing the newest converted value to the ram chip. And we have the computer read the RAM chip at its leisure.

This is just one way the A/D could have been interfaced. This particular converter has a signed four and a half digit resolution. It can, therefore, convert to values from -19999 to +19999, giving it a moderately high resolution.

Reading Multiplexed Data

The following routine may be used to read an A/D interfaced the way we have it in our circuit:

```
999 REM Subroutine to read the A/D
1000 A=10000*PEEK(504)+1000*PEEK(503)+100*PEEK(502)+
     10*PEEK(501)+1*PEEK(500)
1010 A=A*(-1*PEEK(505))
1011 REM Read the sign from A/D and multiply by -1,
1012 REM if necessary to convert to a negative value.
1020 REM 'A' now equals the value read by the A/D converter.
1030 REM Add any adjusting equations here.
1040 REM Print any messages here.
1050 RETURN : REM End of A/D subroutine.
32767 END
```

You should note the following points about this subroutine:

1. You do not have to start the A/D converter. The A/D we used is in the free-run state which means that it is constantly converting.

2. You do not have to stop conversion.

3. You have to read in five bytes of data to acquire the value from the converter, and one byte for the sign. Each byte contains the value of one of the digits. You then multiply by the appropriate power of ten to give the digits their proper decimal positional value.

4. On line 1010 you read in the sign of the number and set A's sign accordingly.

Conveniently, a 1 at location 505 implies a negative sign, and a zero a positive sign, hence when you multiply 'A' by -1 times the value read from location 505, the result is to give A the correct sign.

5. The choice of locations 500 through 505 for the addresses of the A/D was arbitrary. From a software point of view, it could just as well have been any six locations. The A/D converter can be mapped to any unused memory locations not reserved or otherwise disallowed by your microcomputer.

The On-Board Computer

We included a microprocessor in our A/D circuit so that many control decisions may be handled automatically without wasting the time of your host computer. We chose an INS 8073 microprocessor because it is simple to install and use and it includes a tiny BASIC interpreter in the same chip as the microprocessor. With a dedicated microcomputer, more complex applications can be undertaken.

It is often useful to have a program measure many different analog quantities. For example, if you were doing research on how plants respond to different auxins (plant hormones) you might want to have a computer record the variables associated with each plant. The computer can measure the temperature, the pressure, the ambient light, the soil moisture, etc. for each plant.

For this task, the computer might need an A/D converter for each quantity being measured. Or, because the quantities do not need to be read simultaneously, one A/D can be used if it switches from line to line to read each transducer sequentially. An analog switching device such as the INTERSIL IH6116 can switch among 16 different analog signals. It is called a 16 to 1 multiplexer. By sending the chip a four bit code 0000 to 1111 (0 to 15 base ten), one of the sixteen inputs is passed through to the output. You need a software subroutine to control this multiplexer and synchronize its operation with the conversion subroutines. The circuit in Figure 8 shows an INTERSIL IH6116 16-channel multiplexer connected to an INTERSIL ICL7135 four and a half digit A/D chip.

Not only do you not need to read the plant's environmental conditions simultaneously, you also do not need to read them continuously. The temperature can be read every ten minutes or more. You may want to read the soil moisture more often, directly after watering. It would be helpful if the A/D subroutine could read some channels more often than others. It would also be helpful if the frequency could be dynamic. That is, as conditions change, the time between reading channels would also change. For example, at night you may not require a reading of the light intensity as often as during the day. Or, after fertilizing, you might measure nutrients more frequently.

As you can see, there are many variables to keep track of. How often should the A/D read each channel, when should the reading frequency change, what does each channel represent, etc. If you were to have your computer sampling constantly, it would not have much time to process your data. For this reason, we use a dedicated microprocessor to do the sequencing and controlling of the A/D. The sole purpose of this microprocessor is to coordinate the operation of the A/D, the multiplexer, auto ranging of the input, storage of the data and other operations. Your main computer is now free to concentrate on the data itself and not on the collection of data. It is like having an undergraduate collect the data and run the experiments for a senior researcher who deals with the general theory and interprets the results.

A COMPLETE ANALOG DATA ACQUISITION SYSTEM

Figure 9 shows a functional block diagram of the A/D circuit we designed.

In our A/D, we attempted to include all of the features mentioned throughout this paper. Our circuit will fit on a single printed circuit board approximately five inches square. There are roughly twenty integrated circuits used. Most of them are needed only as interfaces. For example, some of them drive the LEDs in the display, others provide synchronous and asynchronous communications capabilities.

When designing this circuit, we tried to follow a few simple guidelines:

- The circuit uses a dual slope integrating A/D to obtain a resolution of four and a half digits. We traded speed for accuracy and economy. Our circuit makes approximately three conversions per second and is relatively inexpensive. The Intersil ICL7135 A/D converter is in the twenty dollar range. High speed flash decoders of lower resolution may cost many hundreds of dollars.

- The circuit operates from a single power supply. We chose a five volt supply because it is almost always available in a digital computer environment. A circuit with only a single supply is simpler to use because it reduces the number of connections that need to be made. Also, single voltage power supplies are cheaper than multi-output models.

- The circuit contains all necessary timing circuitry for its operation. We have a total of three oscillators in the circuit: one for the A/D, another for the real time clock, and a third for the on-board microprocessor. Our circuit is not dependent on the host computer system for timing signals.

- The microprocessor is able to control the complete operation of the circuit, thereby making it completely software controllable. This means that as the user's needs change, so can the data acquisition system by merely changing a few lines of code. Complicated hardware modifications are not necessary.

- The circuit is able to run by itself. In the event of a host computer crash, the circuit can continue running and resume communications when the host computer is up again. The on-board computer has sufficient RAM to hold a brief listing of the events that transpired while the host computer was down. Such a list is made available to the host upon demand.

- Other features of the A/D converter circuit include on-board displays and various communications options. These are described below.

OTHER FEATURES OF THE A/D CONVERTER CIRCUIT

We have included several enhancements in the A/D circuit we have described.

On-board displays

The A/D circuit has a four and a half digit LED display to indicate the value of the channel being read and a two digit display to indicate the channel number. When the computer loads the code of the channel to be read into the 74LS273 eight bit latch, it also writes to the two digit display. This way you always know which channel the board is reading. If a problem ever develops, knowing which channel is being read can be helpful in diagnosing the trouble.

Auto ranging

Sometimes it is convenient to have a converter be able to work with different voltage ranges. Some transducers produce voltages in the tens of volts, others in the hundreds. If you know in advance what voltage range will be used then you can hard wire the appropriate attenuator/amplifier circuit for the channel. But if you don't know what the exact range will be, you can build a circuit to step through different attenuator/amplifier circuits until the input is in the acceptable range for the A/D. Such a circuit should go from attenuation to amplification so that nothing is burned out due to a voltage overload. We have not included an auto ranging circuit in our A/D to keep it simple and to save money, but one could be added if necessary.

Variable Communication Speeds and Protocols

Because the A/D converter circuit is designed to work with a host computer, it must be able to follow standard communication protocols. Our circuit can be run in RS232 serial or handshaking parallel mode. The microprocessor we used is capable of serial communication at up to 4800 baud. The circuit's flexibility permits it to be interfaced to a variety of host computers. We use an INS8255 programmable peripheral interface to provide flexible parallel input and output.

SOFTWARE TO CONTROL THE ON-BOARD MICROPROCESSOR

The software for the dedicated microprocessor must be a bit more complicated than the few subroutines shown above. The following is a sample of a fundamental control program:

```
10 REM This is an example of a simple control program for a
20 REM dedicated microprocessor.  The program assumes there
30 REM is a real-time clock interfaced to the microprocessor.
40 REM First we initialize the variables we will use.
50 DIM V(16) : For storing the current value of each channel.
60 DIM F(16) : Time between successive readings of this channel.
70 DIM T(16) : Time at which the channel is to be read next.
80 FOR I=1 TO 16
90 F(I)=5 : REM Set default time between readings is 5 minutes.
100 GOSUB 2000 : REM Read the clock
110 T(I)=TIME : REM Initialize each to the current time.
120 NEXT I
130 REM
140 REM Begin the infinite loop to update each channel and
150 REM determine which channel needs to be read next by
160 REM searching for the lowest 'read next' time.
170 L=T(1) : C=1 : REM Assume channel one needs reading first.
180 FOR I=2 TO 16
```

```
190 IF T(I)<T(I-1) THEN L=T(I) : C=I : REM  L is lowest time
200 NEXT I : REM C now equals the channel to be read
210 REM Wait until clock and 'read next' time are equal.
220 GOSUB 2000  : REM Go to subroutine to read the clock.
230 IF TIME<L THEN 220
240 GOSUB 1000 : REM Go to subroutine to read the A/D.
250 V(C)=A : REM Update current value stored for the channel.
260 GO TO 170 : REM Go back to line 170 and start again.
270 REM
280 REM End main program.
290 REM
999 REM Subroutine to read the A/D
1000 POKE(507,C-1) : REM Multiplexer numbers 0-15, not 1-16
1010 A=10000*PEEK(504)+1000*PEEK(503)+100*PEEK(502)+
     10*PEEK(501)+1*PEEK(500)
1020 A=A*(-1*PEEK(505)) : REM Read sign from A/D
1030 T(C)=T(C)+F(C)
1040 REM Update the time this channel should next be read.
1050 RETURN : REM End read A/D subroutine
1998 REM
1999 REM Subroutine to read clock
2000 TIME=PEEK(CLOCKADR)
2010 REM You must write a clock reading subroutine
2020 REM specifically for your computer.
2030 RETURN : REM End of read clock subroutine
32767 END
```

Note: The computer we chose to use, the INS8073 by National Semiconductor, does not support arrays in its BASIC interpreter, but by the creative use of memory, similar data structures can be formed. We are not going into the details of the real time clock here. We are, for the the sake of simplicity, assuming complete compatibility in the format of the clock and the variables T(C) and F(C). Of course, this is usually not the case and a set of conversion equations or string manipulations are needed to use the clock in a program.

As you may have noticed, this program only stores each channel's value in an array, V(C). It does not send the data to the main computer. A working program must also have an interface between it and the main computer. There are several ways to do this:

- Have the dedicated microprocessor store the data in memory that is directly accessible by the main computer. In this way, the main computer could get the most recent reading of any channel at any time.
- Have the main computer request the data from the dedicated microprocessor.
- Have the dedicated microprocessor interrupt the main computer whenever the data is determined to require interpretation by the main computer.

For instance, for a given experiment, you may determine that the temperature of a room should be maintained at 25 degrees Celsius. You want to have the A/D measure the temperature in the room and have the computer compensate for changes by turning on and off an electric heater.

You set up one of the channels to read the temperature of the room. As is to be expected, the temperature reading varies. If the dedicated microprocessor has a sophisticated program, it can be 'told' to interrupt the main computer only if the temperature strays more than 4 degrees from your desired temperature. Thus, the main computer does not need to concentrate on the temperature. It can continue processing its tasks and pause to make corrections in the temperature only when they are necessary. This is the most efficient use of the main computer's time. Such a system is called an interrupt driven system because the dedicated microprocessor 'interrupts' the main computer when it requires its attention.

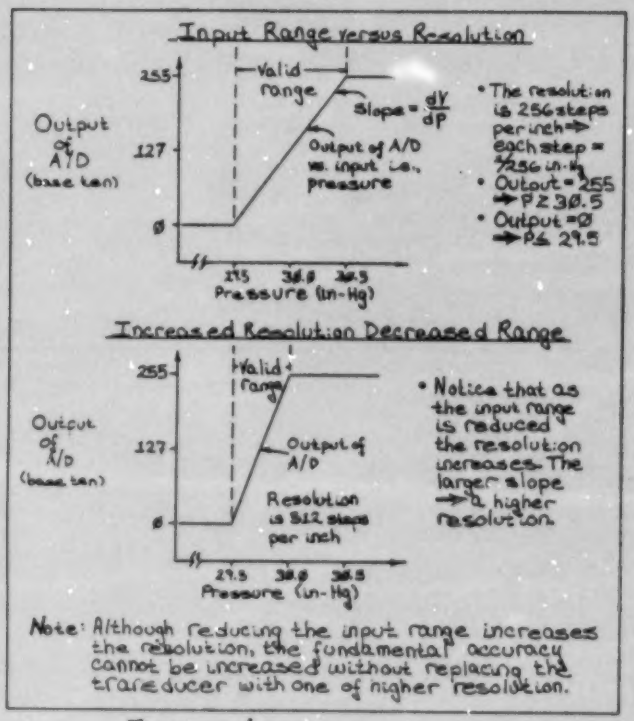
CONCLUSION

Recent advances in technology, have made it relatively easy and inexpensive to use computers in data acquisition. We have illustrated some common techniques used in analog data acquisition systems. By choosing your components wisely, you can optimize the price, speed, and resolution functions.

The uses of A/D converters are practically unlimited. Anything that can be represented as a voltage can be valid input. When an A/D data gathering system is interfaced to a computer control system, a complete feedback loop can be formed. With some careful planning, the system's stability can be made very high. The feedback allows the computer to constantly monitor the results of any action it may take. Such control systems are used throughout industry wherever automation exists and may be adapted for classroom, laboratory, and CAI applications.

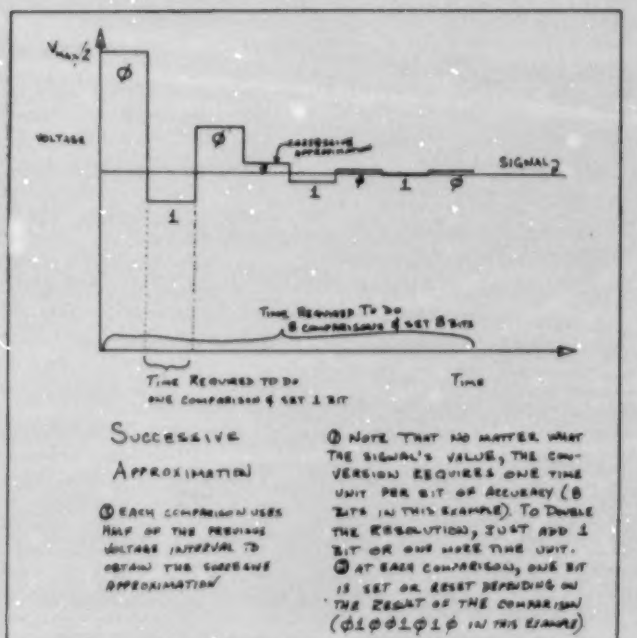
REFERENCE

Tannenbaum, Robert S., and David C. Tannenbaum, "Interfacing Your Microcomputer to the Real World," Proceedings of the 25th International ADCIS Conference., (Bellingham, Washington: ADCIS, 1983) 108-116.


Input Range versus Resolution
Output of A/D (base ten)
255
127
Ø
Valid range
Slope = dV/dP
Output of A/D vs. input abs. pressure
• The resolution is 256 steps per inch ⇒ each step = 1/256 in-Hg
• Output = 255 ⇒ P ≥ 30.5
• Output = Ø ⇒ P ≤ 29.5
29.5
30.0
30.5
Pressure (in-Hg)
Increased Resolution Decreased Range
Valid range
Output of A/D
Resolution is 512 steps per inch
• Notice that as the input range is reduced the resolution increases. The larger slope ⇒ a higher resolution.

Note: Although reducing the input range increases the resolution, the fundamental accuracy cannot be increased without replacing the transducer with one of higher resolution.

FIGURE 1.



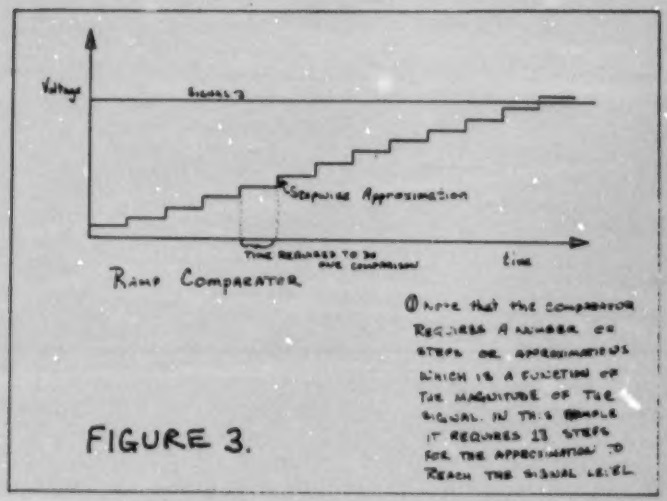
SUCCESSIVE APPROXIMATION

① EACH COMPARISON USES HALF OF THE PREVIOUS VOLTAGE INTERVAL TO OBTAIN THE SUCCESSIVE APPROXIMATION

② NOTE THAT NO MATTER WHAT THE SIGNAL'S VALUE, THE CONVERSION REQUIRES ONE TIME UNIT PER BIT OF ACCURACY (8 BITS IN THIS EXAMPLE). TO DOUBLE THE RESOLUTION, JUST ADD 1 BIT OR ONE MORE TIME UNIT.

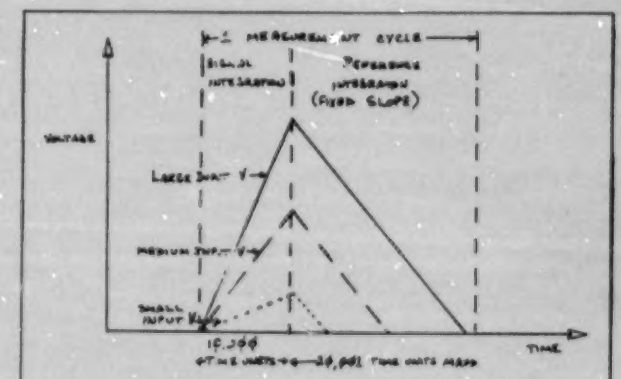
③ AT EACH COMPARISON, ONE BIT IS SET OR RESET DEPENDING ON THE RESULT OF THE COMPARISON (Ø1Ø Ø1Ø1Ø IN THIS EXAMPLE)

FIGURE 2.



① NOTE THAT THE COMPARATOR REQUIRES A NUMBER OF STEPS OR APPROXIMATIONS WHICH IS A FUNCTION OF THE MAGNITUDE OF THE SIGNAL. IN THIS EXAMPLE IT REQUIRES 13 STEPS FOR THE APPROXIMATION TO REACH THE SIGNAL LEVEL.

FIGURE 3.



DUAL SLOPE INTEGRATION

① THE DISCHARGE RATE DURING REFERENCE INTEGRATION IS CONSTANT (FIXED SLOPE), THEREFORE, THE TIME TO DISCHARGE IS DIRECTLY PROPORTIONAL TO THE INPUT VOLTAGE USED TO CHARGE DURING SIGNAL INTEGRATION, JUST COUNT TIME UNITS.

② IF THE CAPACITOR IS DISCHARGED IN LESS THAN 1,000 TIME UNITS (90% OF MAXIMUM TIME) THEN THE INPUT SIGNAL VOLTAGE WAS LOWER RANGE.

③ IF THE INTEGRATING CAPACITOR HAS NOT BEEN FULLY DISCHARGED AFTER REFERENCE INTEGRATION (10,000 TIME UNITS IN THIS EXAMPLE) THEN THE INPUT SIGNAL VOLTAGE WAS OVER RANGE.

FIGURE 4.

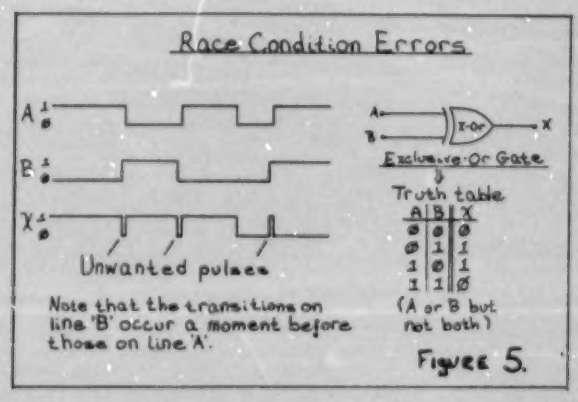


FIGURE 5.

BEST COPY AVAILABLE

Analog to Digital Conversion

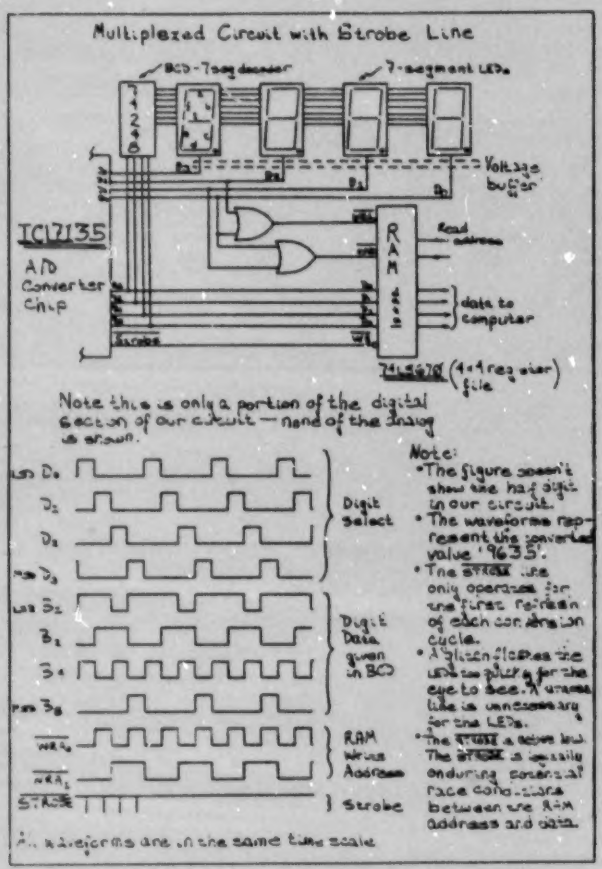

Multiplexed Circuit with Strobe Line
BCD-7seg decoder
7-segment LEDs
Voltage buffer
IC17135
A/D Converter Chip
RAM
Read address
data to computer
Strobe
74LS670 (4×4 register file)
Note this is only a portion of the digital section of our circuit — none of the analog is shown.
Digit Select
Digit Data given in BCD
RAM Write Address
Strobe
All waveforms are on the same time scale
Note:
• The figure doesn't show the half digit in our circuit.
• The waveforms represent the converted value '9635'.
• The STROBE line only operates for the first portion of each conversion cycle.
• A glitch is left in the STROBE to give a pulse for the eye to see. A trailing edge is unnecessary for the LEDs.
• The STROBE is active low. The STROBE is usually enduring potential race conditions between the RAM address and data.

FIGURE 6.

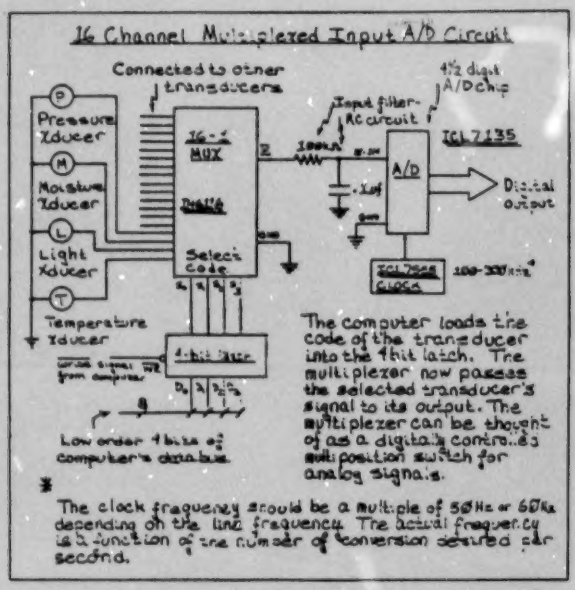

16 Channel Multiplexed Input A/D Circuit
Connected to other transducers
Pressure Xducer
Moisture Xducer
Light Xducer
Temperature Xducer
16-1 MUX
Select code
4½ digit A/D chip
Input filter-RC circuit
ICL7135
A/D
Digital output
ICL7660 CLOCK
100-200 kHz
1-bit latch
Low order 4 bits of computer's data bus
The computer loads the code of the transducer into the 4 bit latch. The multiplexer now passes the selected transducer's signal to its output. The multiplexer can be thought of as a digitally controlled multiposition switch for analog signals.
The clock frequency should be a multiple of 50Hz or 60Hz depending on the line frequency. The actual frequency is a function of the number of conversion desired per second.

FIGURE 8.

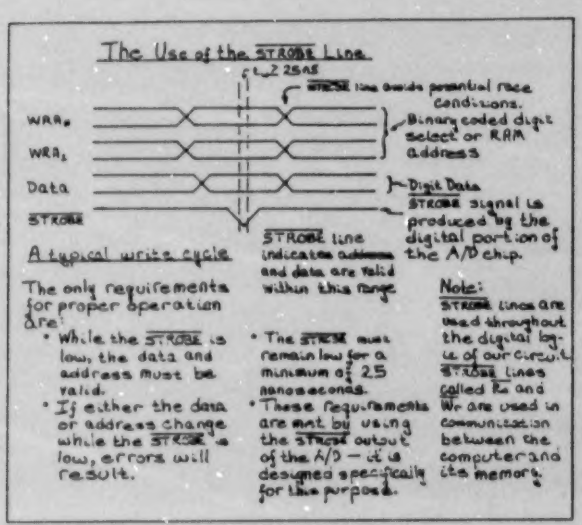

The Use of the STROBE Line
≥ 250 ns
STROBE line avoids potential race conditions.
WRA0
WRA1
Binary coded digit select or RAM address
Data
Digit Data
STROBE
STROBE signal is produced by the digital portion of the A/D chip.
A typical write cycle
STROBE line indicates address and data are valid within this range.
The only requirements for proper operation are:
• While the STROBE is low, the data and address must be valid.
• If either the data or address change while the STROBE is low, errors will result.
• The STROBE must remain low for a minimum of 2.5 nanoseconds.
• These requirements are met by using the STROBE output of the A/D — it is designed specifically for this purpose.
Note:
STROBE lines are used throughout the digital logic of our circuit. STROBE lines called Rd and Wr are used in communication between the computer and its memory.

FIGURE 7.

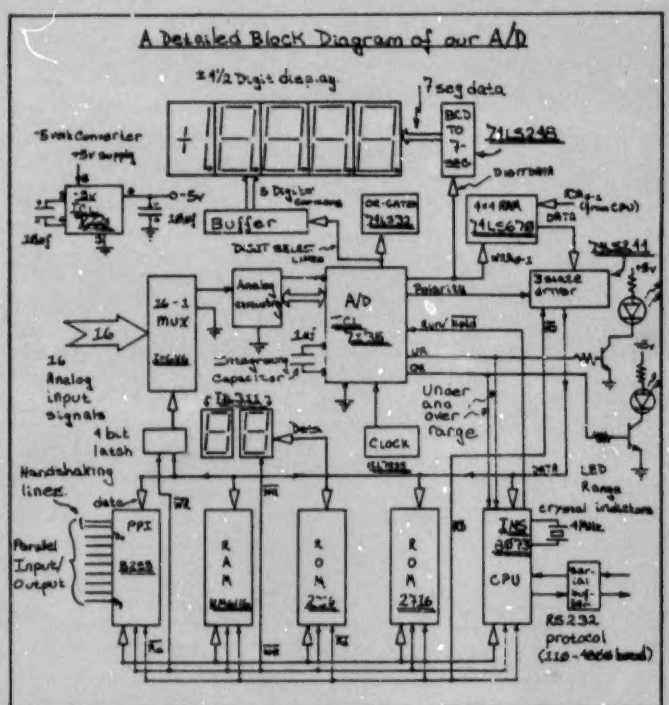


FIGURE 9.

BEST COPY AVAILABLE

Configuring Your Personal Computer

Robert S. Tannenbaum

Hunter College of the City University of New York

ABSTRACT

We are faced with many choices and decisions when we purchase new hardware and software. I have developed some guidelines to help you make these choices and decisions. These are listed and explained in this paper. To illustrate the use of the guidelines, I describe an IBM PC system with numerous peripherals and extensive software which I recently selected and purchased.

INTRODUCTION

Recently I spent a considerable amount of time and money configuring and purchasing a new microcomputer system. I realized that the choices I made were specific to my needs and preferences, but the process I followed in making my decisions was a general one and may be of interest to others. I decided to present the guidelines which I try to adhere to whenever I configure or add to a system. These constitute the bulk of this paper. As an example of the application of these guidelines, I have included a brief description of the system I put together.

CHOOSING SOFTWARE AND HARDWARE -- SOME GUIDELINES

You should buy ready-made software and hardware whenever you can rather than writing your own programs or designing and building your own boards. But choosing among the many available peripherals and software packages is an important and difficult task. Software and hardware packages designed for similar purposes are not necessarily equal and they do not have equal support, documentation, and flexibility. I will discuss several fundamental questions to which you should have satisfactory answers before buying a particular package.

Purpose

What are your needs and how well does this product meet them? You must first make a complete and honest assessment of what you really need. It is too easy say, "I want to be able to do everything." No product can "do everything" and you probably don't need to. Unless you accurately describe what you need to do, you will probably buy a product which does more than necessary. You will pay for it in increased purchase price and inefficiency of operation because you will be carrying software or hardware overhead which you never use.

If you analyze your needs, you will be able to evaluate whether the product meets them. For example, if you are in the market for a text processor and you plan to be writing papers with it which include footnotes, you know that you must check how each package permits and facilitates footnoting. Or, if you are looking for a spreadsheet accounting system and you know that you need at most 50 columns and 150 rows, you have a guideline for judging the acceptability of various systems based on the size of the spreadsheets they can handle. Or, if you are planning to buy a multi-function board and you know you will want to interface a coordinate digitizer which connects to a game port, you will not consider a board without such a port.

Resource Requirements

What other hardware and software does this product require and do you have it? You may have found a package which seems to meet your needs, but will it run on your system? Do you have enough memory, or the color graphics capability, or the disk capacity, or operating system which it requires?

For example, you may have found a database management system which looks attractive, but it requires at least 64k more RAM than you have on your system now. You must include the cost of adding 64k of memory to your system in your decision about whether to purchase this software.

Documentation

Is the documentation adequate and well written so that you can install and use the product yourself? Poor documentation can be the source of frustration and hours of wasted time. You probably should not consider purchasing any product that does not have a complete manual which includes:

- Installation instructions
- A general tutorial on the product
- Detailed explanations of how to use all of the features
- Numerous pertinent examples
- Help in handling problems
- A thorough index.

Support

Who stands behind this product ready to help you in case of difficulty? Buying a piece of software or equipment may not be quite as expensive as buying a new car, but it is analogous in many ways. You should carefully check the service department before you buy a new car. And you should look for computer companies and retail outlets which have a reputation for responding with helpful solutions to user's problems and which are known to correct bugs and update their products with new and improved versions. If you buy by mail you may get a reduced purchase price, but you will probably get little or no support or service in case of problems.

Reputation

What have been the experiences of others and what are their reviews of this product? The car buying analogy is appropriate here again. You wouldn't buy a new car without talking to as many people as you can who own the model you are interested in and you would read as many reviews as possible in reputable magazines. You should do the same with hardware and software. Read the reviews in such magazines as *Byte, PC, PC World,* and *Creative Computing* before making your decision. And, talk to people who already own the item. If you don't know anyone who has the package you are looking at, use the bulletin board or the forums of one of the networks to find someone; or attend some meetings of a local computer club and ask. Before I decided to choose Turbo Pascal, PC-File III, PC-CALC, the Sweet-P plotter, and the Amdek 710 monitor, for example, I read numerous reviews, checked several forums, and talked to friends and club members.

Flexibility and Growth

Can this product grow or adapt and change as your needs change? You realize that your needs will change with time. It is not always easy to predict exactly what changes will occur, but you should have a general idea of what you will need in the next year or two. If possible, you should consider a package which can grow to meet those needs as well as today's.

This may be the most difficult question to deal with in choosing a product because you can not be sure of your future needs. But you must remember that you are making a large investment when you choose a software package or a piece of hardware. The initial price is only a small portion of the cost. You will be devoting considerable time to using the system and the files which you produce or the peripheral you connect may be compatible only with that package and nothing else. If your needs change enough so that you must change software or hardware at a later date, all those files may have to be significantly modified or even rewritten and the peripheral may have to be replaced.

Trial Run

Can you try out the product to see if it really meets your needs? You wouldn't buy a car without a test drive and you shouldn't buy software or hardware without a trial run. Many computer stores will let you test a product in the store before you buy it. Be sure that it is you doing the testing, not some salesperson giving you a demonstration with canned examples.

Arrange with the store for a hardware system identical to yours to see how all of the features of the new package will function on your system. Bring a realistic sample of your work to the store and spend enough time (several hours at least) working on your project so that you get a thorough feeling for what the product can and cannot do.

Alternatively, some software firms now offer sample diskettes for a very modest price ($5-10) which you may use on your own system at your convenience to review the product. This approach does not let you test all of the features of the software, but does let you try it on your hardware with your data.

Price

How much does it cost? Are there various versions of the product with different prices? And is it worth it to you? You must know how much you have to spend and whether you can afford this product. But even if you can afford it, you must be sure that it is worth the investment. Perhaps a less expensive package will meet your needs satisfactorily and leave you resources to purchase something else in addition.

Be sure that you shop around. Many computer products are available at significant discounts from different stores, clubs, and mail order suppliers. The discounts sometimes range as high as 50%, so do not consider the list price, but rather the best price you can find and the services mentioned above when you decide whether it is worth purchasing.

APPLYING THE GUIDELINES -- MY NEW SYSTEM

I am in the process of writing a textbook on the applications of computers in the humanities and the social sciences. I had decided during the earliest stages of planning that the book would use the IBM PC as its standard whenever a specific, hardware-dependent example was needed. Therefore, the general family of microcomputer was a given, but the specific model and all of the peripherals and software had to be chosen.

I needed a system which would serve two main purposes:

- It would be my text processor on which I would write and edit the book.
- It would be my test and demonstration system on which I would try out various humanities and social sciences computer applications and produce and verify the examples which are included in the book.

These requirements dictated a rather large and flexible system. You may argue with any or all of my choices and, given my experience with the system, I might make slightly different choices today. But overall I am quite satisfied with my selections.

Computer

Given my large text processing, storage, and flexibility needs, I decided that I had to use the IBM-XT. If I were purchasing today with the same needs, I would get the IBM-AT, but that was not announced when I purchased my system. There will always be a newer and better model tomorrow. But you need your computer today, so buy the best system available and don't second guess yourself.

I shopped around to find the best deal. I found that by driving about 200 miles round trip (about $50) I could save over $400 dollars in taxes and I could have the system immediately off the shelf. It was worth the few hours of shopping.

Memory and Multi-Function Boards

I had decided that I needed all the memory I could get and several special-purpose and multi-function boards because of the size and range of the applications I planned. Again I shopped around. I found that the best prices were available from a discount mail order firm in Chicago. Since I felt confident that I could handle installation and maintenance without a dealer, I chose to pay the lower purchase price. If you feel you may need dealer support, pay the few dollars more at purchase time.

I bought enough memory to bring the system to full capacity (640k). I bought a Quadram multifunction board because it did everything I needed, was functionally equivalent to the other available boards, and was a few dollars less expensive. I also bought a Corvus Mirror system board which allows me to back up my hard disk on a video cassette recorder.

Printer

I thought long and hard about which printer to buy. I knew that I was writing a book and would need to print many, many pages of draft and final copy. I also thought I would like to have beautiful, letter-quality output. And I needed a printer which could be driven by the text processing and formatting software I wanted to use. I compromised. I chose not to spend $2,000 or more for a high-speed, letter-quality printer (for which I would probably have had to write my own specialized driver software). I settled on the Epson FX 80 which is fast enough and flexible enough for my needs, my software drives it, and it cost only about $500. It is a little awkward to change from text to graphics and intersperse the two (such as to draw boxes around figures), but I have learned to live with that. This paper was printed on my Epson.

Configuring Your Personal Computer

Monitors and Adaptors

My XT came with a monochrome monitor which is excellent for text processing and programming. I also needed a color graphics monitor for charts and graphs and statistical and artistic applications. I decided that I needed a high resolution monitor to be able to do the artistic graphics I wanted to write about.

I looked at over a hundred monitors at an exposition in New York City and decided that the Amdek 710 was the one I wanted. It combined high resolution (640 x 480 pixels) and sharp colors at a price I could afford (just over $500). But it was not available on the market yet. What I had seen was a prototype model specially prepared for the exposition. Since I was not planning to work on color graphics immediately, I decided to place my order and wait. I am glad I did. It took almost six months, but now that I have been using the monitor for a month, I am very satisfied with it. If you can afford to wait, sometimes it pays.

Once I had chosen a color monitor with 640 x 480 resolution, there was only one choice for the adaptor board -- the Tecmar Graphics Master. Tecmar is the only company with a PC adaptor board for this range of color resolution.

Tecmar's software is quite powerful, but it is not totally compatible with all other software. I cannot run both my Quadmaster and my Tecmar software simultaneously and I have yet to figure out how to access the Tecmar high resolution color commands from Turbo Pascal. Therefore, I use Tecmar and Quadmaster at different times and I do only medium resolution graphics in Pascal; I do high resolution graphics in other languages.

I decided to install the 8087 mathematics co-processor for improved performance in number-crunching and complex graphics computations. Several of my software packages are significantly speeded by this chip. I shopped around and found that I could save a significant amount if I purchased the chip by mail. No matter where I bought it, I would have to install the chip myself (a very simple matter) and there is no servicing necessary (just replacement if it is faulty), therefore, there was no reason not to purchase by mail.

Hardcopy Graphics Input and Output

Many applications in the humanities and the social sciences require hardcopy graphics and graphic input of data. So I needed a plotter and a coordinate digitizer.

Plotters range widely in size, number of features, and price. I am only doing demonstrations and trying out various graphic ideas, not producing thousands of working drawings, so I chose a medium speed, medium resolution, low-priced plotter -- the Sweet-P Six-Shooter. It handles two sizes of paper, holds six pens, plots quickly and accurately, has both serial and parallel interfaces, and uses its own expanded or the HP standard graphics commands. I have been very pleased with its performance.

I may have made a slight mistake when I purchased only the parallel cable for connecting my plotter to the computer. I chose that connection because I had an available parallel port and no available serial port. But that connection permits only output *to* the plotter. With a serial RS232C connection I would have been able to read *from* the plotter as well as write to it; to ask the plotter what the current pen position is and proceed accordingly. I will replace the parallel connection with a serial one if I decide to do a lot of plotting.

For graphic input, I chose the VersaWriter. It has an arm with potentiometers at the "shoulder" and "elbow" joints. You place the arm at the point(s) you want to digitize. The VersaWriter's software reads the potentiometer settings and converts them to the digital coordinates of your point(s). It is simple, inexpensive, and does an accurate job. It can be used to input the coordinates of any two-dimensional drawing for research, business, or art. The resulting picture may be stored, edited, displayed on your monitor, or printed.

Software

Explaining all of my software options and choices would take far more space than I have here. I will simply list the major pieces of software I chose with a very brief comment on each.

- DOS 2.1 Operating System -- The obvious operating system for the PC. Always use the latest available version of the operating system.
- Turbo Pascal -- If you are using Pascal, this is the fastest and least expensive compiler. It is not the most powerful and the documentation is terrible.
- IBM Professional Editor -- A very powerful and flexible editor for large documents (such as chapters of a book). It does not do much formatting for printing -- it is not a "what you see is what you get" editor -- but you can edit any type of text. I put control tags in

my text to control formatting via Script-PC.

- Script-PC -- An excellent package for formatting and printing text according to IBM publication standards. It uses the same controls as IBM Script (DCF) on IBM main frame computers.
- Word Proof -- A spelling checker which is very fast, easy to use, and has a large dictionary.
- IBM Logo -- A good implementation of an excellent language for education and graphics.
- PC-File III -- An inexpensive database management system which will handle small to moderate sized databases very efficiently and is excellent for teaching the fundamentals of database management.
- PC-Calc -- An inexpensive spreadsheet system for small to moderate sized spreadsheets. It is excellent for teaching the basics of spreadsheet systems.

CONCLUSION

I have listed some guidelines which you should follow when buying a new system or adding to an existing one. I have described the system which I recently purchased as an example of applying these guidelines. The guidelines may be summarized as follows:

Don't buy the product unless you have satisfactory answers to all of the questions I have raised.

- What are your needs and how well does this product meet them?
- What other hardware and software does this product require and do you have it or will you have to buy it?
- Is the documentation adequate and well written so that you can install and use the product by yourself?
- Who stands behind this product ready to help you in case of difficulty?
- What have been the experiences of others and what are their reviews of this product?
- Can this product grow/adapt/change as your needs change?
- Did the product meet all of your needs and expectations when you made a trial run using it?
- How much do the various versions of the product cost and is it worth it to you?

No product will be perfect, but if you ask these questions and base your decision on the answers, you should be able to find the product which best meets your needs.

BEST COPY AVAILABLE

A MACHINE-INDEPENDENT DATA STRUCTURE FOR THE REPRESENTATION OF MUSICAL CONCEPTS AND STRUCTURES: COMPUTER-GENERATED MUSICAL EXAMPLES FOR CBI

J. Timothy Kolosick
School of Music
University of Arizona
Tucson, Arizona, 85721
(602) 621-7018

ABSTRACT

The elegance of any program can be judged, at least partially, by the method and reasoning guiding the placement of data into data structures and the ease with which the various procedures and functions operate on those data. One programming goal should be to organize data in such a fashion that the relationships between the various data are represented by that data's relative positioning in the structure. Such a structure can allow the data to be easily manipulated in order to produce information based on the data and their relationships.

This paper proposes such a data structure for representing musical pitch relationships. Although the entire field of music can not be represented at this point, the proposed pitch data structure allows the development of additional programming modules for the generation and recognition of any tonal or atonal collection of tones by ordering pitches (data) in such a way as to make possible the calculation and representation of musical structures (information) through the use of standard arithmetic formulae. This paper will include a discussion of the application of this data structure to musical structure generation and transposition thereof for CBI lessons in music.

The heart of any computer program is its group of data structures. As Niklaus Wirth writes, "Data structures and algorithms...are the basic elements of every computer program. The choice of data structures and the design of procedures to manipulate them hold the key to verifying that a program does what it is meant to do." (Wirth, 1984) Indeed, the elegance of a program can be judged, at least partially, by the method and reasoning guiding the placement of data into the structures and the ease with which the various procedures and functions operate on those data. One programming goal should be to organize data in such a fashion that the relationships between the various data are represented by that data's relative positioning in the structure. Such a structure can allow the data to be easily manipulated in order to to produce information based on the data and their relationships.

Much of the computer's time in CBI applications is dedicated to the presentation of materials, that is, visual and aural stimuli prompting the student to respond and, through continued activity, to learn from work with the machine. The materials to be presented are generally organized as a series of fixed items, specific structures which are presented sequentially or selected at random. However, if we accept as our goal the representation of instructional elements in such a way as to show their relationships with one another, we will need to develop more sophisticated data structures than those presently used to represent only text, questions, and answers. The representation of an entire field of study, or even a portion of it, in one group of data structures would allow the machine to appear to

BEST COPY AVAILABLE

understand that area of interest and be able to generate much of the material for questions, answers, and other student/machine interaction. This arouses both fascination and fear in the minds of educators and elicits the familiar cry, "it can't be done! Can it?"

This paper proposes such a data structure which allows the machine to "understand" relationships present in the basic units of musical structure. Although the entire field of music can not be represented at this point, the proposed data structure allows the development of additional programming modules for the generation and recognition of any tonal or atonal collection of pitches and allows relationships to be discovered, analyzed, and further demonstrated by the machine. It orders pitches (data) in such a way as to allow the calculation and thereby the representation of musical structures (information) through the use of standard arithmetic formulae. This paper will deal with one application of these capabilities, that is, "musical structure generation" and transposition thereof for use in CBI lessons in music. "Musical structure generation" is perhaps more appropriate to describe this activity rather than the term "composition." Certainly this stage of development will lead to higher levels of operation with this data representation, enabling researchers to develop new algorithms for combining structures into musical works and arrive at a better understanding of the compositional process and its computational model. Recognition and understanding of a composer's methods can be developed through music analysis, recognizing how certain pitch choices move through such an all-purpose data structure. But for now, let us consider the methods for generating fundamental musical elements and structures.

THE PITCH REPRESENTATION

The suggested representation of pitch is a series of variable sized nodes containing the data necessary to represent particular notes. Each node, called a record, contains the data necessary to represent the particular qualities of each note which give it its identity and peculiar spelling. Thirty-five records are necessary to represent all note names. The contents and arrangement of the data within each record would be variable according to the desires and interests of the programmer. Each datum is stored in a discrete portion of each record called a field. The data stored in each record keep the same field assignments and the fields keep the same position in the record throughout the array. This allows each field to be associated with a particular meaning within each record. A record might be arranged as shown in Figure 1.

The presence of the proper spelling of a pitch (e.g. C♯) allows one to work with the actual note name rather than using "enharmonically blind" pitch classes in order to represent a note. A reference to pitch class is also included in Figure 1, however this could be calculated based on the ordering method used in this representation. The base frequency is the frequency of the note in the lowest octave to be used in the representation, the presence of which would allow a programmer to investigate the generation of examples using various tunings of early music. Since all western tuning systems have pure octaves, one can represent the frequencies for one octave and calculate the frequencies for all other octaves. Values for musical sound sources could allow the pitches to be played and, like unto the frequency information, one octave could be represented and all others derived using an octave value added to the represented value in the record. Any remaining fields could be used to represent elements to be associated with each particular note name or certain fields listed above could be removed. Advantages to using such an array of records include the expandability and memory efficiency of the system as any number of attributes associated with a particular note can be stored once in one group of memory locations.

ORDERING OF THE PITCH RECORDS

The array is set up in such a way that the records are arranged in a series of fifths cycles beginning on Fbb as shown in Figure 2. This arrangement sets up a particular intervallic relationship between the records as one progresses through the array. The fifths are, of course, all perfect fifths. The perfect fifths ordering allows the computer to have all of the basic intervallic, harmonic, melodic, and

Field 0	Field 1
Field 2	Field 3
Field 4	Field 5
Field 6	Field 7

Spelling	Pitch Class
Base Frequency	Value for Sound Source
Key Number (Piano)	etc...

Figure 1

F♭ C♭ G♭ D♭ A♭ E♭ B♭ F♭ C♭ G♭ D♭ A♭ E♭ B♭ F♮ C♮ G♮ D♮ A♮ E♮ B♮ F♯ C♯ G♯ D♯ A♯ E♯ B♯ F𝄪 C𝄪 G𝄪 D𝄪 A𝄪 E𝄪 B𝄪

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34

Figure 2

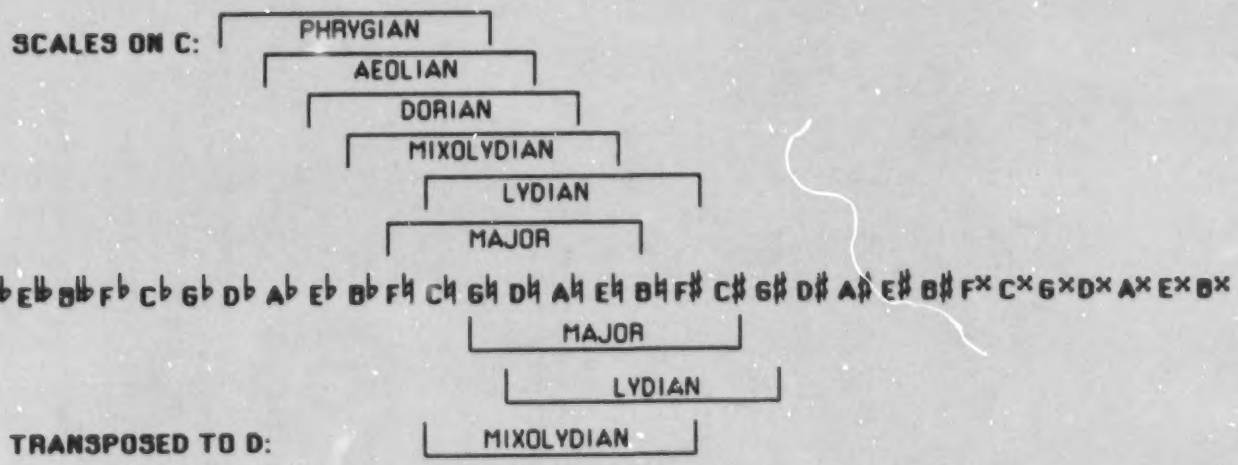

SCALES ON C:
PHRYGIAN
AEOLIAN
DORIAN
MIXOLYDIAN
LYDIAN
MAJOR
F♭ C♭ G♭ D♭ A♭ E♭ B♭ F♭ C♭ G♭ D♭ A♭ E♭ B♭ F♮ C♮ G♮ D♮ A♮ E♮ B♮ F♯ C♯ G♯ D♯ A♯ E♯ B♯ F𝄪 C𝄪 G𝄪 D𝄪 A𝄪 E𝄪 B𝄪
MAJOR
LYDIAN
TRANSPOSED TO D:
MIXOLYDIAN

Figure 3

set-theoretic elements at its disposal. All elements of any style of western music can be represented, transposed, and compared in both compositional and analytical applications. The principle employed is the notion of the "offset" which is a basic concept of many computer programming tasks. The ordering of the records in cycles of fifths, as shown, allows that if two or more notes share a particular musical relationship, the offsets or distances between those notes as found in the array can be moved to another portion of that array yielding the same musical relationships between the resulting notes. Some examples follow.

SCALES

Within the representation all scales and modes are present in the form of adjacent records, that is, any 7 adjacent records will yield a scale. Figure 3 shows examples of these. Within each group of records forming a scale, one of the records represents the tonic note as shown in the following table.

MODE	TONIC
Lydian	First Record
Ionian (Major) . . .	Second Record
Mixolydian	Third Record
Dorian	Fourth Record
Aeolian (Minor). . .	Fifth Record
Phrygian	Sixth Record
Locrian.	Seventh Record

Table 1

As groups of 7 records are viewed throughout the array, all scale degrees keep their same placement within each scale and proper note spellings prevail. For instance, the record number of a leading tone is always greater by 5 than that of its tonic. The record number of a subdominant of a key is always less by one than that of its tonic. Record numbers of all chromatically raised scale steps are greater than the values of the 7 scale records while all lowered scale degrees are lower in value.

Generating notes for specific scale degrees can be accomplished by a simple formula and a small array of offsets shown in Figure 4.

ARRAY OF OFFSETS: (Major Scale)

Indices:	0	1	2	3	4	5	6
Offsets:	0	2	4	-1	1	3	5

Figure 4

The record number of a given scale degree is calculated by adding the value found in the ARRAY OF OFFSETS to the value of the tonic, i.e. the record number of the tonic note in the "Perfect Fifths Array" (P5ARRAY). This is shown in the formula:

NEW NOTE = TONIC RECORD + ARRAY OF OFFSETS [SCALE DEGREE]

A simple loop from 0 to 6 (representing scale degrees 1 through 7) would establish the names of all the notes of a particular scale in ascending order. The last remaining problem to be solved is the adjustment of the octave placement of those notes. The notes in the P5ARRAY are all represented in the same, that is, lowest octave available on a particular sound source. The octave adjustments can be accomplished by comparing the pitch level of each note of the scale with that of the tonic. If any note is not higher than the tonic, its register would be adjusted one octave higher. This assures the pitches to be in ascending order. The entire scale can then be transposed to the octave desired for presentation to the student.

INTERVALS

Like unto scales, all intervals are inherent within the array's structure and can be calculated forming transpositions of identical intervals properly spelled. The structure allows the use of proper tonal labels to identify intervallic relationships rather than depending on interval class or "number of half-steps" designations. The intervals are determined by an offset in the P5ARRAY. The offset refers to the distance between the first note and the second note of a given interval. For example, the offset between C and D is 2 because moving two P5ARRAY locations to the right from C will yield the note D. Generating intervals and the problems of octave

placement will be discussed later in this paper. The relationships between offsets and particular intervals is found in the following table.

OFFSET	DOWN	UP	INTERVAL CLASS
-12	Aug7	dim2	0
-11	Aug3	dim6	5
-10	Aug6	dim3	2
-9	Aug2	dim7	3
-8	Aug5	dim4	4
-7	Aug1	dim8	1
-6	dim5	dim5	6
-5	Maj7	min2	1
-4	Maj3	min6	4
-3	Maj6	min3	3
-2	Maj2	min7	2
-1	P5	P4	5
0	P8/P1	P8/P1	0
1	P4	P5	5
2	min7	Maj2	2
3	min3	Maj6	3
4	min6	Maj3	4
5	min2	Maj7	1
6	Aug4	Aug4	6
7	dim8	Aug1	1
8	dim4	Aug5	4
9	dim7	Aug2	3
10	dim3	Aug6	2
11	dim6	Aug3	5
12	dim2	Aug7	0

(+12 = raised enharmonic spelling.)
(-12 = lowered enharmonic spelling.)

Table 2

CHORDS AND SETS

Chords and other sets of pitches can be represented by a series of offsets within the array, one offset for each interval present between members of the chord. Some examples of tonal structures are shown in Figure 5. The pitches are calculated by finding the distance between pitch nodes in the P5ARRAY as indicated by values found in an array of offsets for each chord or set type. Again, each group of offsets can be shifted throughout the array yielding the same structure transposed by the interval of the offset of the shift. (See Table 2 for intervallic equivalents of offsets.) Proper tonal spellings automatically result in every case.

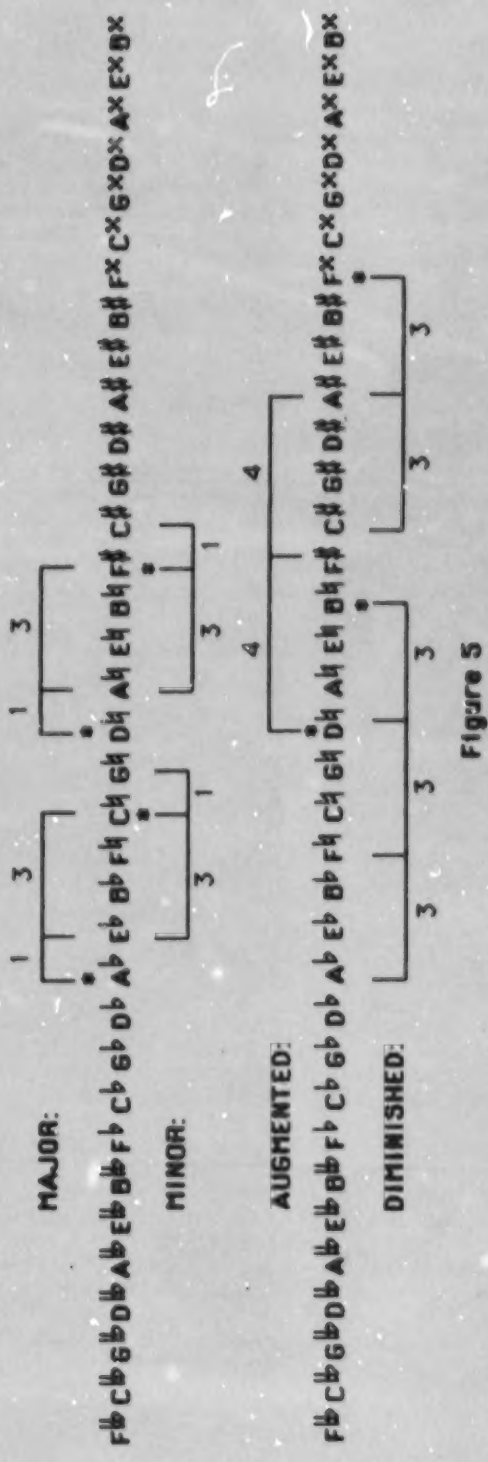


Figure 5

The programmer could use the indices shown in Figure 5 from "left to right" in order to locate the proper pitches for a particular structure, e.g. a minor triad would be represented by [3,1]. This would entail finding a starting pitch, then moving to the right 3 records for the second pitch and then further one record for the third. Using this approach, it would be necessary to designate one note in the group as the root of the chord. Such notes are marked by an asterisk (*) in Figure 5. Perhaps a more succesful approach would be to refer to a chord as a group of offsets from its root, as shown in Figure 6, using the order of its notes from the root upward in the chord's closest position.

ARRAY OF OFFSETS FOR CHORDS:

Quality	Root	Third	Fifth
Major	0	4	1
Minor	0	-3	1
Diminished	0	-3	-6

Figure 6

Knowing the relationship between offset and the chord member (Root, Third, or Fifth) allows for octave adjustments to be easily made. Like unto scales, if the lowest intended pitch (e.g. root for root position triads) is not lower than any other pitch, the octaves for each pitch will be adjusted according to the octave desired for presentation.

COMPUTER-GENERATED INTERVALS

The method for interval generation to be discussed in detail here, is based on the use of a group of assembly language routines which produces graphics and sound using any Apple II series computer and the Temporal Acuity Products' digital-to-analog converter (DAC). This collection of programs, known as the Musician's Toolbox was written by Paul Daley, J. Timothy Kolosick, and Michael W. Turner and it will soon be made available through a software publisher. Its purpose is to provide music educators and programmers with a set of fast, memory-efficient routines in order to develop instructional, compositional, or analytical materials for their own schools and/or offer courses in such programming activities.

The score representation for the Musician's Toolbox is a series of numbers in groups of 2 or 3 bytes (based on programmer's choice) for each pitch. The first, or rhythm, byte contains information as to a particular note's duration, its stem direction, presence of tie or dot, and whether or not the accidental associated with the note's spelling should be displayed. The second, or pitch, byte contains pitch information and will be discussed shortly. The third, or display, byte, if used, controls the amount of space between simultaneities and the distance between the note head and its displayed accidental. Default values are used consistently if no display byte is present.

Except for the displaying of particular accidentals, the rhythm or display bytes will not be changed in a standard interval drill scenario. Our concentration here will be on the calculation and manipulation of the contents of the pitch byte information. The pitch byte is divided as shown in Figure 7 in order to represent the pitches to be played.

The "right" 6 bits are used as an offset, or distance, from the beginning of a frequency look-up table. This gives one up to 63 values, as the condition "0" is used to indicate a rest. The Temporal Acuity Products DAC has a range of 4 octaves or 49 notes. The two remaining bits are used to represent the spelling of the note associated with the sounding frequency. Since any given pitch has no more than 3 possible spellings, these values 0,1, and 2 can be represented in 2 bits. The conditions are associated with a note's lowest, middle and highest spellings.

Two additional data structures are needed in the program used to call or execute the Musician's Toolbox routines in order to generate intervals. The first is the "Perfect Fifths Array" as described earlier (P5ARRAY). For this application, the only data needed for each pitch is the individual spelling for each note name and the frequency table offset for one octave. These two records used in the structure discussed earlier correspond to the left 2 bits and the right 6 bits of the byte respectively. The lowest playable octave on the music board (DAC) was chosen and by a process of adding multiples of 12 to the number representing the frequncy table offset, various octaves can be

Figure 7

	m2	M2	m3	M3	P4	P5	m6	M6	m7	M7	P8
Offset, up	-5	2	-3	4	-1	1	-4	3	-2	5	0
Offset, dn	5	-2	3	-4	1	-1	4	-3	2	-5	0
1/2 steps, up	1	2	3	4	5	7	8	9	10	11	12
1/2 steps, dn	-1	-2	-3	-4	-5	-7	-8	-9	-10	-11	-12

This array allows any number of intervals to be chosen from.
The name of each interval would be stored in a string array.

Figure 8

calculated. For example, if the representation of the frequency offset for D is 3, then the D which is two octaves higher could be calculated by D + (Octave * 12) or 3 + (2 * 12) = 27.

The second data structure, a two-dimensional array (INFO.ARRAY), contains the necessary information for the intervals to be presented in any particular drill, that is, it holds the offsets in the P5ARRAY (as shown in Table 2 earlier) and the number of 1/2 steps present in each interval (See Figure 8). These numbers may seem quite redundant as shown but their placement requires very little additional memory and allows much faster execution, always welcome in microcomputer applications. Of course, any group of individual intervals could be represented in such an INFO.ARRAY and used in such a drill.

With these data structures in place, the following steps could be performed by the computer in order to generate randomly-selected intervals based on randomly-selected pitches.

1. Randomly select an integer between 1 and the number of possible intervals, i.e. select an interval (INTERVAL).

2. Randomly select an integer referring to a location in the P5ARRAY between F flat and B sharp inclusive, i.e. select a beginning note (FIRST). This limitation will prevent mistakenly choosing intervals whose resulting pitches do not exist, e.g. the pitch a major seventh above A double sharp would be G triple sharp.

3. Randomly select an octave for the first note. Due to the range limitations of 4 octaves the second and third octaves are always chosen. This way one need not double check to see if the second note is out of range if one is dealing with simple, as opposed to compound intervals (OCTAVE).

4. Choose a direction for the interval, up or down (UPDOWN).

5. Get the value of the first pitch by the following formula.

PITCH1 = P5ARRAY[FIRST] + (OCTAVE * 12)

6. Calculate the frequency portion of the second pitch (PITCH2) as follows.

   PITCH2 = PITCH1 + INFO.ARRAY[INTERVAL,NUMSTEPS+UPDOWN]

   This adds a positive or negative number of half steps (NUMSTEPS) to the frequency look-up value of PITCH1's record.

7. Replace the spelling portion of PITCH2 with the number found in the formula below.

   INFO.ARRAY [INTERVAL,OFFSET+UPDOWN]

   At this location the correct spelling is found in the spelling portion of the record in the P5ARRAY.

8. With the combination of spelling values and the frequency look-up tables the score representation for PITCH2 is complete.

9. With most interval drills the accidental for all notes other than those in their natural state will be displayed. By comparing both FIRST and the location of the spelling for PITCH2 with the group of locations in the P5ARRAY one can determine the need for an accidental. If one of the values is less than 14 (F natural) or greater than 20 (B natural), the resulting notes will need the accidental displayed.

This discussion of a musical pitch organization can only suffice as an introduction and does not, in any way, reveal its full potential. This data structure has been used in several computer programs for music instruction and has been found to be effective and accurate. The Music Sound and Graphics Package (MSGP) by Paul E. Dworak, North Texas State University, uses the P5ARRAY to translate a scale degree based representation of a musical score into sound and graphics. This allows the user to represent one score which is easily transposed by the MSGP. The Musician's Toolbox, using a specific pitch representation in the score, uses the P5ARRAY in order to decode the spelling used in the two highest bits of the pitch byte. The P5ARRAY was also used for generation of musical examples in the Jazz Dictator. This program "composes" musical chord progressions in a standard jazz style for the development of aural skills in this area of study.

Hopefully more work will be done using this basic data structure for musical applications in CBI, analysis, and composition. With more applications and further investigation of the P5ARRAY and supporting data structures, its use can be refined and its potential fully realized.

Reference: Wirth, Niklaus. "Data Structures and Algorithms." Scientific American, September, 1984, pp. 60-69.

PROVIDING ACCESS TO COMPUTER-BASED EDUCATION IN FREE PUBLIC LIBRARIES

Clella B. Murray
University of Delaware

ABSTRACT

This paper addresses the implementation and potential of testing CBE programs at a public library. The potential is accessibility to special groups, families and lifelong learning projects, not easily tested in an educational environment. The implementation of the one-year pilot program funded by the Office of Computer-Based Instruction (OCBI) at the University of Delaware involved decisions on types of software and hardware, preparation of librarians, and curriculum for the general public. The results show a sampling of testing information that can be obtained.

Introduction

This presentation discusses developmental testing of CBE programs at a public library. Because of its unique nature the public library offers a fertile field for testing computerized instruction on two groups, families and adults. Information found from these studies would not only be valuable to the educational community but would have economic implications, as the studies would apply to the new emphasis in education (adult or lifelong learning) and the fast growing home computer market.

Why Outreach to Family and Adult Learning Groups?

The prediction is that there will be a personal computer in about one out of three American households by 1986 and in eight of ten households by the year 2000 (Dreyfuss, 1984). Many parents will purchase these microcomputers to help their children with homework, or reinforce what a parent or teacher sees as a weakness in a child's academic background. Some parents will go through the programs with the children. Yet how few studies have been done on family educational computer usage compared to the wealth of information on games. Dreyfuss (1984) states that to have more control of the market, software-game producers will be writing for the whole family, with more emphasis on the adult consumer; many companies view educational software as a sleeping giant.

There have been references in studies and research to the socio-economic divisions in computer use between the rich and poor, black and white. Affluent white schools teach more programming and have more computers than poorer black schools, where computing generally focuses on drills and reading (Campbell, 1984). Could one reason for this be that white children from affluent homes are entering school with computer experience gained in their homes? What better place than the free public libraries to test young readers as members of a family and at the same time to provide computer usage to the underprivileged?

Adult education is the new direction of many schools, universities, and communities. It is anticipated that college enrollment in the next twenty years will remain steady, but the proportion of eighteen to twenty-four year olds will decline and the number of older undergraduates will increase (Fauquet, 1983). Educational institutions that do not prepare for this shift in enrollment may be forced to close, according to Fauquet (1983).

Computer-assisted instruction is tailor made for adult education. As noted by Vacc (1984), "computer-assisted instruction provides immediate feedback, permits self-selection lessons, and is non-threatening." In addition the use of computer-management programs makes it possible for educators untrained in CBE to

individualize programs and supervise the adult's progress (Fahy, 1984). There are computer tools available to study and to aid adult learners. Vacc (1984) feels the word processor would help adult learners develop writing skills. Meyer (1982) referred to twenty-five million adults who are disabled readers. Microcomputer programs provide reading help in sentence-combining activities or word lists, and the microcomputer may be used as a tachistoscope to control the speed of word presentation for adult readers (Vacc, 1984). Yet with the plethora of published articles on computer-assisted instruction, few articles deal with the adult-learner (Vacc, 1984). Slaninka (1983) deplored the lack of research on development of adult writing skills.

There are several explanations for the lack of information on adult learners. One is a reticence of adults to use computer terminals. Larson (1980) lists the causes of this reticence as four barriers: informational, situational, institutional, and attitudinal. The University of Texas nursing graduate course, nicknamed "Conquering Computer Cowardice," addresses overcoming this reticence problem (Grobe, 1984). The strategies used at Texas could be adapted to other disciplines. A second explanation for lack of information is the deficiency of places where a cross section of adults are available for testing. The public library is the perfect testing ground.

Why Are Public Libraries Good Places to Collect Data?

The public library is free to all persons of all ages; there is no discrimination based on economics, age, or sex. It is open in the evenings as well as during the day, so working persons can avail themselves of its services. Libraries offer a physical plant with heating and (in many) air-conditioning. Librarians are trained professionals accustomed to working with the public, giving help and instruction, and making reference searches. Libraries are friendly, familiar places where adults are not likely to be embarrassed and children feel at home. But perhaps most important, libraries have a tradition as a source of learning and a history of service to adult learners (Smith, 1984). Now libraries can build on their traditional rolls and offer to their patrons a new opportunity, access to educational lessons on computers.

How is a Library-Testing Program Implemented?

In 1983, Dr. James Morrison, of the College of Human Resources, University of Delaware, and Ms. Yvonne Puffer, of the Newark Free Library, received a grant from the Office of Computer-Based Instruction (O.C.B.I.) at the University of Delaware and the Friends of the Newark Free Library to place one terminal in the Newark Free Library. The Control Data Corporation (CDC) IST-1 terminal was to remain for one year giving the general public free access to the PLATO® system on the University of Delaware CYBER® 174 mainframe. The terminal was installed in July, and training of the librarians began in August. Ten librarians were trained to assist library patrons.

Use of PLATO System

The advantages of using the PLATO system as opposed to using microcomputers were carefully considered. For some situations, microcomputers may be more desirable, but in this instance the PLATO system was chosen on the basis of the following features: (1) minimal use of librarian time, (2) hardware and software security, (3) data collecting facilities, (4) large numbers of available programs, and (5) grant possibilities from the University of Delaware.

As is shown in Table 1, the use of the mainframe meant the librarians' time would not be involved in (1) reviewing or cataloging programs to determine which programs to present to the public, (2) explaining the disk drives or handling the diskettes, or (3) supervising the security of the diskettes, disk drives, or the terminal. The largest single item in a library's budget is the salaried staff (salaries for the Newark Library were $295,090 out of a $401,643 budget for 1983-1984), so this was the single greatest consideration.

Anyone who has had contact with a public library will recognize the toll destruction and theft have on the library resources. The mainframe use answered the physical security needs. In a public library, portability would have been desirable in that the terminal could have been moved to the appropriate section, and children or adults would not have intruded into the other's areas and created noise problems. However, the security of a fixed location prevailed in the final decision.

The router programs on the PLATO system have great data collection possibilities available. With microcomputers it is necessary to write a program to collect data and to have additional disk drives to run the data collection.

From the menu driven selection system at the University of Delaware a large number of programs is available. To build a catalog

---

PLATO® is the registered trademark and service mark of Control Data Corporation.
CYBER® is the registered trademark and service mark of Control Data Corporation.

Table 1. Comparison of Microcomputer and Mainframe Features

	MICRO	CYBER 174 MAINFRAME
Cost	Cheap to buy computer	Expensive to buy terminal
	Library must purchase diskettes	Expensive to access
		Must pay for phone line
Software	Library must buy, review, and catalog diskettes	Access to large number of programs
	Librarian's time to supervise	Do not need to review or catalog programs
		Little librarian supervision
Security (Physical)	Easy theft of diskettes, disk drives, and computer	Few security problems
	Diskettes easily destroyed or damaged	
Location	Portable	Must be near phone jack
Versatility	Could use as word processor or with spread sheets	Could not readily use as word processor or with spread sheets
User friendly	Possible trouble with disk drives for new user	Very user friendly
Collect data	Difficult	Good data collecting facilities
Source of hardware & software	Rental, purchase, or free for advertising computer stores	OCBI grant procedure

of microcomputer programs each program must be reviewed, chosen, procured, and cataloged separately.

With all of these advantages it was decided to apply for funding from OCBI to use a terminal that accesses the programs on PLATO system.

Librarian Preparation

Some of the librarians were older persons who had never had computer experience and it was extremely important to them to have a text to follow until they became familiar with the machine. A great deal of time was spent preparing for librarian use a manual that was simple and unthreatening. The end product was a series of screen prints similar to the one shown in Figure 1, with handwritten notes giving instructions. The handwritten notes accomplished two purposes, they destroyed the sacredness of the screen image and they encouraged users to take their own notes and become physically involved. Movement within the computer program was indicated in the manual index page by hand drawn arrows pointing to the appropriate plastic divider as shown in Figure 2. The librarians at first followed the screen displays page by page in the book. In the last half hour of the last instructional session, three fourths of the users never opened their books; the need for the security blanket was over.

The classes, lasting for two two-hour sessions, were given in a CBE classroom at the University of Delaware. The librarians were given instructor sign-ons. The first session was a practice of signing on and off the computer and exercises that would make them feel at home with the terminal. As Boss (1984) points out, "The greater the

BEST COPY AVAILABLE

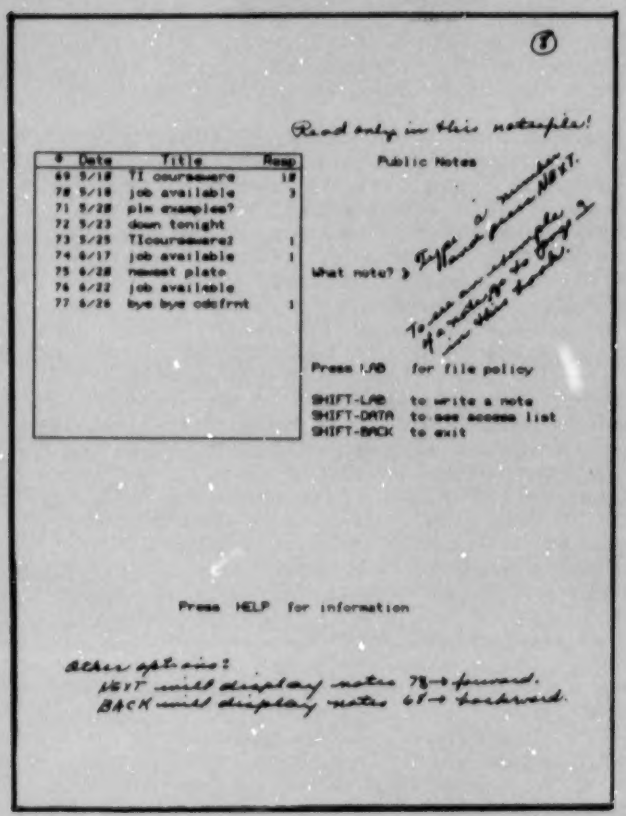

Read only in this notesfile!
Public Notes
Date Title Resp
69 5/18 TI courseware 18
70 5/18 job available 3
71 5/28 pln examples?
72 5/23 down tonight
73 5/25 TIcourseware2 1
74 6/17 job available 1
75 6/28 nearest plato
76 6/22 job available
77 6/26 bye bye cdnfrmt 1
What note? >
Press LAB for file policy
SHIFT-LAB to write a note
SHIFT-DATA to see access list
SHIFT-BACK to exit
Press HELP for information
other options?
NEXT will display notes 78→ forward.
BACK will display notes 68→ backward.

Figure 1. Page in Librarians' manual showing written directions.

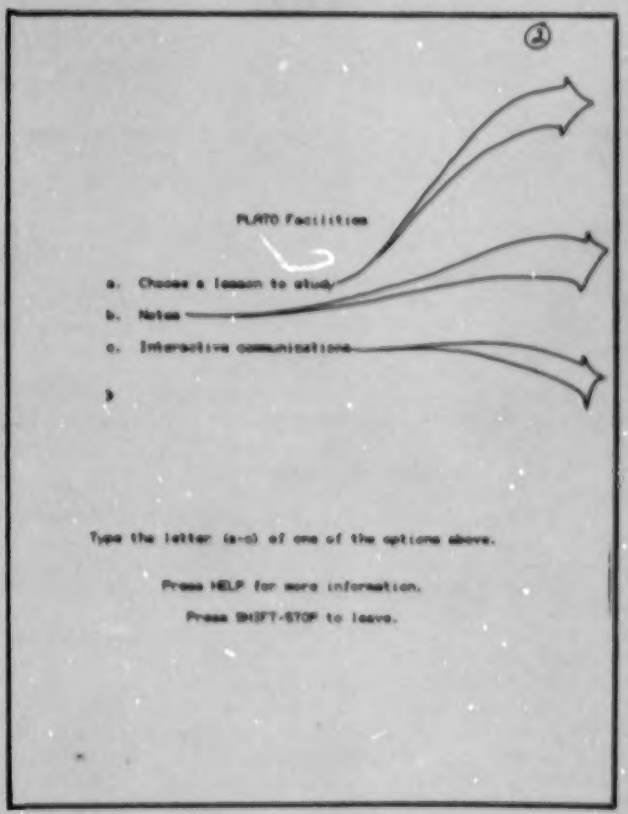

PLATO facilities
a. Choose a lesson to study
b. Notes
c. Interactive communications
>
Type the letter (a-c) of one of the options above.
Press HELP for more information.
Press SHIFT-STOP to leave.

Figure 2. Index page of librarians' manual indicating direction options.

discomfort of the user with the terminal the greater the resistance to the technology." The exercises consisted of using the editor to write personal notes or enter notes in notesfiles and to using the TERM key to make comments or talk. In the second session the librarians signed on as library patrons and learned to move about in the indexes. The class grew from feeling comfortable with the terminal to enthusiasm for computers and a feeling of self-confidence. One sixty-three year old, who confided she had been awake two nights worrying what she would do when library patrons approached her for help, made the boast, "Let them come!" The librarians imparted their enthusiasm to the library patrons.

Curriculum/Curriculum Access

The adult curriculum consisted of twenty-one alphabetically listed disciplines. The subjects were chosen as those having the greatest appeal to the general population. For the adult sign-on, three entries were chosen for each discipline. The first two were sample programs, the third was a general index to each subject. For the youth sign-on the entries were divided into appropriate grade levels, corresponding to the subjects studied in the schools. Each person using a youth sign-on was encouraged to try other levels. Only seventeen programs and one index were advertised on the youth sign-on. On the introductory page of each sign-on, the first entry was "How to Use PLATO" by Jessica Weissman. (Both youth and adult sign-ons are multiple sign-ons. No individual student sign-ons were used. An average of 600 persons use the library per day. It would be impossible to create a sign-on for each patron.) Also available to the patrons was a read-only notesfile where librarians announced programs and special activities. Any questions from programs via TERM comment were recopied and answered in the general notesfile.

Security

The security of a mainframe terminal is much simpler than that of the microcomputer and the accompanying diskettes. As Savage (1984) suggested, the best security is to have a plan ahead of time to deal with a situation. The only precaution

BEST COPY AVAILABLE

necessary in this experiment was to place the computer within eye-view of the main desk to prevent vandalism. In the fourteen months of use no vandalism to the terminal had occurred; yet during the same time span the Newark Library lost $18,000 (or 1164 items) from a total book budget of $58,913.

Publicity

The terminal installation was publicized in the local newspapers and radio stations, in the Friends of the Library Newsletter, and the City of Newark Cicular. An open house was held at the library on a Sunday afternoon to answer any questions. The library prepared handouts and posters.

Results

Of the results three were surprising. First, probably little publicity was needed. The terminal was being used fifty percent of the available time, before any publicity got under way. People were so eager to use the computer, it was difficult to keep them from signing on before the librarians were trained. Second, monitoring terminal use required little of the librarians' time. It had been expected that supervision would be a limiting factor, since most libraries operate on a rigid salary budget. Third, the fact that there was only one terminal was not a difficulty, even though the Newark Library serviced 181,504 persons in 1983-84 (one-year period). Terminal time was reserved one hour per week. Reservations were made in person with no phone reservations accepted. If there were an unreserved time, it was filled on a first-come basis.

Table 2 is an example of the kind of information that can be easily obtained. This table shows usage during the thirteen month period of September 1, 1983 to September 30, 1984. The youth and adult sign-ons used 2932 hours on the system. Keep in mind that youth and adult refer only to the name of a sign-on not necessarily to the age of the person using the computer. Anyone under eighteen was encouraged to start with a youth-signon. However, many youth sign-ons were families with young children or adults who felt they needed remedial work, while some of the adult sign-ons were precocious school children.

Adults used 679 hours total and youth used 2253 hours total on the system. On adult displays 44 programs and 20 indexes were advertised; there were 5503 uses of 468 programs with a 6.1 minute average. On the youth display 17 programs were listed and one index (math); there were 6675 uses of 79 programs with 14.0 minute average use.

It is notsworthy that youth hours on the system were over three times adult hours, although there were three times as many programs advertised for adults and the accesses were comparable. A study of the programs accessed by youth shows that the most frequently used programs were those with number and word game formats and beginning typing. The accesses to math and word games were comparable. The indexes adults accessed the most frequently were business, career planning, computer science, and art. The most frequently accessed subjects were beginning typing, astronomy, insects, basic programming, and Spanish.

The twelve librarians spent a total of 89.6 hours on the system, an average of 7.5 hours. They entered 122 separate individual programs or indexes. The average hours per sign-on were 1.9. Apparently the librarians felt their role was that of a guide not a user.

Table 2. Usage Statistics of the Multiple Sign-ons, Youth and Adult, and the Instructor Sign-on

PLATO System Usage at Newark Free Library September 1, 1983 to September 30, 1984			
	ADULT	YOUTH	LIBRARIAN
Total time on system	679 hrs.	2253 hrs.	89.6 hrs.
Advertised programs and/or indexes	64	18	80
Programs used	468	79	122
Accesses to programs	5503	6675	318
Average use	6.1 min.	14 min.	1.9 min.

Conclusion

The overall total was eighty-four percent usage of the library terminal during the sixty-one hours per week that it was available. On the basis of this pilot test the county librarian is writing proposals for outside funding (1) to purchase similar terminals for each of the four county libraries and (2) to cover hook-up fees for three years.

With this overwhelming public interest, it is hoped both educational institutions and computer companies will consider the potential of the public library as a testing ground for future studies. The side effects of public library testing might be to reduce the computer isolation of our elderly, a growing segment of our population, and to expose computer-assisted instruction to non-affluent families.

References

Boss, R. W. (1984). Technology and the modern library. Library Journal, 109, 1183-89.

Campbell, P. B. (1984). The computer revolution: Guess who's left out? Interracial Books for Children BULLETIN, 3, pp. 3-6.

Dreyfuss, J. (1984, April 2). What will send computers home. Fortune, pp. 71-2.

Fahy, P. J. (1984). Learning about computerized instruction with adults: One school's trials, errors and successes. Educational Technology, 24, 11-16.

Fauquet, T. W. (1983, September). Recruiting and retraining older students: Institutional and faculty roles. Kansas State University Center for Faculty Evaluation and Development, Idea Paper No. 10.

Grobe, S. J. (1984). Conquering computer cowardice. Journal of Nursing Education, 23, 232-239.

Larson, G. A. (1980). Reaching hard-to-reach adults: Overcoming barriers to communication. New Directions for Continuing Education, 8, pp. 27-38.

Meyer, V. (1982, May). Teaching reading as a search for meaning: Helping the whole student. Paper presented at the International Reading Association Convention, Chicago, IL.

Savage, N. (1984, April). Facing up to library security. Wilson Library Bulletin, pp. 562-564.

Slaninka, S. (1983). Writing as a learning process - implications for the adult educator. Lifelong Learning: The Adult Years, 6, pp. 15-16.

Smith, J. C. (1984, April). The public library: a history of service to adult learners. Lifelong Learning: The Adult Years, 7, p. 2.

Vacc, N. N. (1984, April). Computers in adult education. Lifelong Learning: The Adult Years, 7, pp. 26-28.

THE COMPUTER AS TRANSITION FROM OBJECT MANIPULATION TO ABSTRACT VISUO-SPATIAL THOUGHT

Donna F. Berlin The Ohio State University-Newark

Arthur L. White The Ohio State University-Columbus

ABSTRACT

This study explores the learning model which suggests that a concept is acquired first through manipulation of concrete objects followed by transformation of the concrete objects into semi-concrete representations, followed by internalization of the concept through abstract representations.

Microcomputer simulations of mathematics manipulative activities were used to determine how children differ in their understanding of mathematics concepts when using the simulations compared to using the concrete materials, or a combination of simulations and concrete materials.

The sample for the study included a total of 113 children distributed according to male and female and 2nd and 4th grade level. The treatment condition consisted of three levels based upon the proportion of activities using concrete objects to those using computer simulations (concrete-only, concrete and computer, computer-only).

The criterion measures assessed the children's ability to: recognize and duplicate a design, recognize and extend a pattern, and locate objects in space.

The results indicated that:

1. Fourth graders performed better than second graders.
2. Rural, white children performed better than suburban, black children with activities involving the computer.
3. Rural, white girls performed better than suburban, black girls using concrete-only activities.
4. Rural, white boys using activities involving the computer performed better than suburban, black girls using concrete-only activities.

Introduction and Rationale

The National Council of Teachers of Mathematics states as one of its priorities for the 1980's ". . . the use of electronic tools such as calculators and computers should be integrated into the core mathematics curriculum." With the increasing demand for computer use in the classroom, its unique role in terms of the learning process and individual cognitive style needs to be examined.

Mathematics educators have generally accepted a theory of concept development based upon a progression from concrete manipulation of objects, to representational form, and finally abstract thought. It is suggested that children acquire concepts first through concrete experiences with manipulative objects, extend their understanding of the concept by the transformation of the concrete objects into semi-concrete representations (i.e., pictures, images) and internalize the concepts through abstract thinking (i.e., symbols, letters and numbers). This model has been associated with Piagetian theory and the development of cognition through stages marked by concrete, semi-concrete (imagery), and abstract actions. Similarly, Bruner (1960) defines three systems of information-processing to represent the world. The systems involve the enactive (actions and appropriate motor responses), iconic (images), and finally, the symbolic (language). There has been much literature to support the importance of the initial, concrete manipulation of objects (Dienes, 1971; Gagné, 1970) and the capability for abstract thought is obviously a desired end. However, the transition between these two stages needs to be explored. The use of the microcomputer as an interactive link between the concrete and abstract levels of thinking is suggested.It should be noted that while Billings (1983) makes a case for the use of the microcomputer during the initial stage of concept development based upon the computer's interactivity, graphics, motion capabilities, and storage features, the computer provides an image or representation which is not a concrete object. For this reason, the use of the computer seems more appropriate during the intermediate stage of concept development. This research focuses specifically on the effect of microcomputer graphics simulations as the intermediate step in the learning process.

Statement of the Problem

The problem in this study was to investigate the effect of interactive microcomputer graphics simulation activities on the transition from concrete manipulations of objects to abstract thinking in elementary school mathematics.

Objectives

Specifically, the study will assess student performance:

1. In recognition and duplication of designs.
2. In recognition and extension of patterns through
   a. Seriation and
   b. Symmetry
3. In spatial orientation and discrimination tasks (i.e., location of objects in space).

METHOD

Subjects

The study involved 113 elementary school children randomly selected from two Ohio elementary schools. The sample was drawn so as to include 2nd and 4th grade boys and girls as shown in Table 1.

Table 1
Site by Sex by Grade Level Distribution of Subjects

	Site 1		Site 2	
Grade	Boys	Girls	Boys	Girls
2nd	13	15	15	15
4th	15	14	11	15

Treatment

The treatment conditions varied on two dimensions. See Table 2.

Dimension I described the proportion of manipulative activities (concrete) to microcomputer graphics simulation activities (semi-concrete).

There were three levels of this dimension:

a. Concrete manipulative activities only.
b. Combination of concrete manipulative and computer graphics simulation activities.

c. Computer graphics simulation activities only.

Dimension II described the socio-cultural background of the student populations in the elementary schools. There were two levels of this dimension:

a. Predominately black, suburban (Site 1).
b. Predominately white, rural (Site 2).

Table 2
Distribution of Subjects by Treatment Conditions and Socio-Cultural Background

		Dimension II Socio-Cultural Background					
			Site 1		Site 2		
Dimension I		Sex/ Grade	Boys	Girls	Boys	Girls	Total
Manipulative Objects to Computer Ratio	Manipulative Objects only	2nd	4	5	5	5	19
		4th	5	5	4	5	19
	Manipulative Objects and Computer	2nd	5	5	5	5	20
		4th	5	4	3	5	17
	Computer only	2nd	4	5	5	5	19
		4th	5	5	4	5	19
	Total		28	29	26	30	113

Students were randomly assigned by grade level and gender to the treatment conditions (Dimension I). The students received mathematics instruction using either pegboards and colored cubes and/or computer simulations of these materials. There were a total of twenty tasks which developed the following mathematical concepts and skills:

a. Recognition and duplication of design
b. Recognition and extension of pattern
c. Location of objects in space

The teachers used these activities over a three-week period in a manner determined and prescribed by the research team.

At the end of the treatment period, two paper-and-pencil instruments requiring reflective abstract thought were administered to all subjects. Test1 requires pattern recognition; Test2 requires pattern extension and orientation. Scores on these instruments comprised the dependent variables.

Design and Analysis

The data analysis design for this study is a 3 x 2 x 2 x 2 - 4 way-multivariate analysis of variance with two dependent variables. Follow-up procedures for interpretation of the results include use of the standardized discriminate functions for any significant effects, the inspection of the univariate effects and the correlations of each dependent variable with the discriminant functions. For interpretation of interaction effects the means for the dependent variables were plotted and a post hoc procedure employed to determine specific differences. Differences at or beyond the .05 level were considered statistically significant.

These statistics were used to test differences in mathematics concept development for the following:

a. between use of concrete manipulative materials and computer graphics simulations.
b. between levels of socio-cultural backgrounds.
c. between 2nd and 4th graders.
d. between boys and girls.
e. differential effects of the treatment conditions related to site, gender, and/or grade level.

RESULTS

Table 3 reports the results of the multivariate analysis of variance and univariate analyses of variance for all significant main and interaction effects. Inspection of Table 3 reveals a significant main effect for both site and grade level on both criterion instruments and significant interaction effects for site by grade level and site by treatment by sex for only Test2.

Table 3
Analysis of Variance For Site, Treatment, Sex, and Grade Level Significant Effects for Performance on Two Spatial Tests

SOURCE	df	MS	F	p
Site x Treatment x Sex	4,176		1.74	.14
Test1	2,89	.20	.16	.86
Test2	2,89	10.05	3.40	.04
Site x Grade	2,88		10.44	.001
Test1	1,89	.004	.003	.96
Test2	1,89	60.027	20.38	.001
Grade	2,88		7.68	.001
Test1	1,89	17.92	13.76	.001
Test2	1,89	12.37	4.18	.04
Site	2,88		7.62	.001
Test1	1,89	12.91	9.91	.002
Test2	1,89	25.34	8.57	.004

Table 4 shows the means and standard deviations for both criterion instruments for subjects classified by site, sex, and grade level assigned to each of the three treatment groups.

These summary statistics indicate that fourth graders performed better than second graders on both criterion instruments. The mean performance of students at site 2 (rural, white) was higher than at site 1 (suburban, black) on both Test1 and Test2.

Means and Standard Deviations for Performance on Test1 and Test2 by Site, Treatment, Sex, and Grade Level

Dimension I		Dimension II: Socio-Cultural Background								
			Site 1				Site 2			
		Sex	Boys		Girls		Boys		Girls	
		Grade	Test1 M (SD)	Test2 M (SD)	Test1 M (SD)	Test2 M (SD)	Test1 M (SD)	Test2 M (SD)	Test1 M (SD)	Test2 M (SD)
Manipulative Objects to Computer Ratio	Manipulative Objects Only	2nd	4.5 (1.29)	4.0 (1.41)	2.8 (1.64)	1.6 (1.52)	3.2 (1.30)	1.4 (1.52)	3.6 (0.55)	2.6 (1.95)
		4th	4.0 (1.00)	1.4 (2.07)	4.2 (1.30)	1.2 (2.17)	5.0 (1.41)	2.5 (1.73)	4.8 (1.30)	4.2 (1.79)
	Manipulative Objects and Computer	2nd	3.2 (1.30)	2.6 (1.82)	3.2 (0.84)	2.6 (2.19)	4.0 (0.71)	1.8 (1.30)	5.2 (0.84)	2.2 (1.64)
		4th	4.8 (1.64)	1.6 (1.82)	4.75 (0.96)	1.25 (1.89)	4.67 (1.56)	6.0 (0.00)	5.4 (0.55)	3.8 (0.84)
	Computer Only	2nd	4.25 (1.26)	2.25 (2.87)	3.2 (0.84)	1.8 (1.30)	4.8 (1.30)	2.8 (2.05)	4.2 (1.10)	1.2 (1.64)
		4th	4.0 (1.41)	2.0 (1.23)	3.8 (1.10)	2.4 (2.19)	4.75 (0.50)	4.75 (0.50)	5.0 (1.0)	4.0 (1.0)

The Scheffé post hoc multiple comparisons for differences between the means of the Site by Sex by Treatment interaction effect for Test2 resulted in the following significant ($p \leq .10$) differences:

1. Rural, white females did better than suburban, black females when using concrete-only materials.
2. Rural, white males did better when using the concrete-computer or computer-only activities than did the suburban, black females using concrete-only materials.

It can readily be seen in Figure 1 that the major contribution to the significant three-way interaction is the positive slope of the line from concrete-only to concrete-computer for rural, white males.

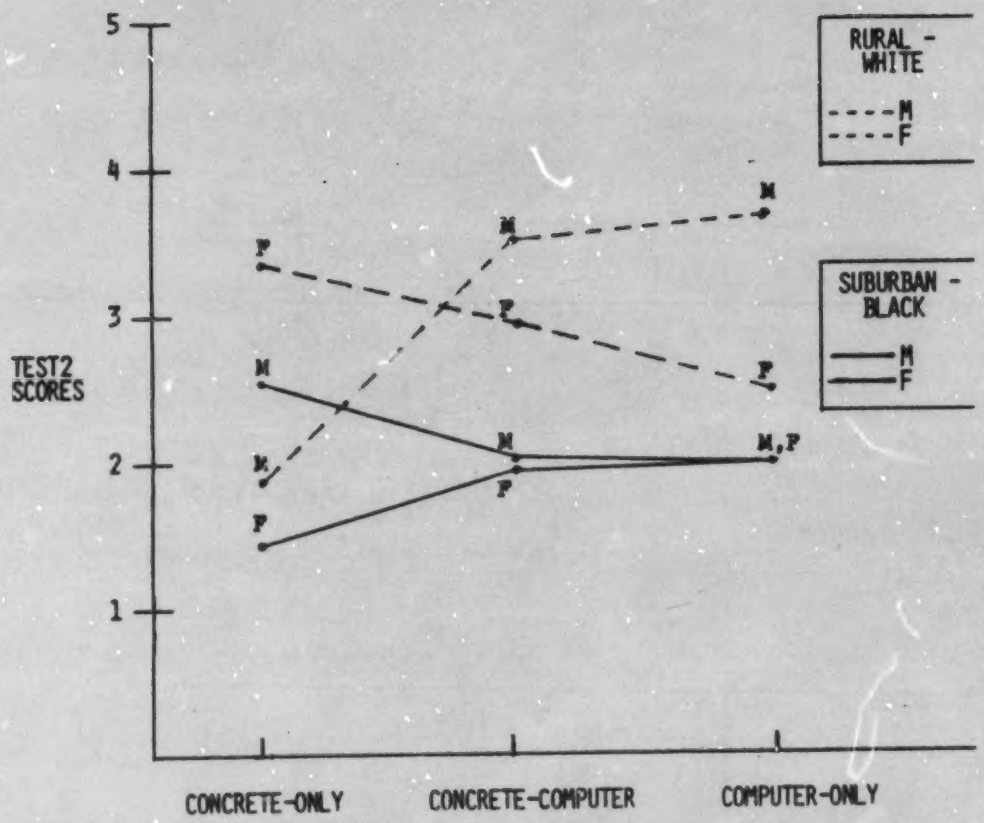


Figure 1
Site X Sex X Treatment Interaction Effect

The slopes for rural, white females and suburban, black males from concrete-only to concrete-computer are both negative. Males in the rural, white site significantly benefit when computer activities are a part of instruction. The other groups either show no effect or perhaps a detrimental effect when computer activities are used.

Discussion

It is clear that the responses to the criteria are a function of grade level. Fourth grade children are better able to recognize, orient, and extend designs and sequence patterns. It is also apparent that geographic, socio-economic, cultural, and/or sex factors are related to children's performance on the criterion measures. The results would suggest that it is important to consider these characteristics before the selection of the concrete or semi-concrete instructional modes.

Further research for purposes of better defining the nature of these influences on the effectiveness of concrete, manipulative materials and computers on abstract thinking are in progress. Attempts will be made to increase the probability of creating differences and the probability of finding differences if they exist. The steps under consideration include:

1. Larger sample.
2. Integrating concrete and computer activities throughout the concrete-computer treatment.
3. Using more activities.
4. Using activities over a longer period of time.

These modifications along with replications in a variety of socio-cultural settings are necessary. The researchers have plans to replicate this work in Mexico, Costa Rica, and Trinidad, as well as in the United States (Ohio) so as to explore other racial, cultural, language and socio-economic groups.

In addition, research exploring other technology such as calculators and the influence on the transition from concrete manipulation to abstract understanding are being planned. This type of research may serve to provide another embodiment of the transition construct and guidance for future curriculum development.

References

Billings, K. (1983). Developing mathematical concepts with microcomputer activities. Arithmetic Teacher, 30, 18-19, 57-58.

Bruner, J.S. (1960). The process of education. New York: Vintage Books.

Dienes, Z.P. (1971). Building up mathematics. London: Hutchinson Education.

Gagné, R.M. (1970). The conditions of learning. New York: Holt, Rinehart & Winston.

National Council of Teachers of Mathematics. (1980). An agenda for action: Recommendations for school mathematics of the 1980's. Reston, Virginia: The National Council of Teachers of Mathematics.

Piaget, J. & Inhelder, B. (1969). The psychology of the child. New York: Basic Books.

INTELLIGENT TUTORING OF PHYSICS VIA PERSONAL COMPUTER--EMPHASIS ON LEARNING

Stephen F. Brown, James Miller,

Kathleen Swigger, North Texas State

ABSTRACT

With personal computers becoming faster and having larger memories, artificial intelligence will soon emerge as the driving force behind intelligent computer-assisted instruction (ICAI) in the educational environment. It is during these developmental years that the basic framework and methodology of intelligent tutoring will be founded. One of the major areas of investigation and development will be in learning theory. In this paper we describe a process for incorporating an expert's knowledge of introductory non-calculus physics (statics and kinetics) into a representation within a personal computer which can run LISP, the premier language of artificial intelligence. The learning theory of Ausubel is used as the basic guide for instruction. Information is organized by the expert in such a way that the student only learns material which has meaning relative to existing knowledge. Following the learning hierarchy developed by Gagne, the expert knowledge of physics is decomposed and classified into types of learning ranked according to complexity. Starting with specific problem-types, the construction of the expert system begins with a depth-first identification of prerequisite types of learning (i.e., principles, concepts, discrimination, verbal information). After a few problem-types have been incorporated into the organizer, the initially hierarchical organization becomes an extremely networked hierarchy (a net-archy). Since the progression of instruction through this net-archy may follow any path with meaning, and the amount of learning at each node will be path-dependent, Bloom's taxonomy for assessing cognitive behaviors is used to provide a measure of learning. The Bloom-level score at each node is used to determine the direction of future instruction, although the student is allowed to view his/her position within the organizer to allow the course of instruction to reflect the motivation of the student. Finally, a process by which the expert system can be provided with the rules of physics is given. This forms the basis for facilitating expert problem generation, BUGGY identification, an intelligent explanation facility, and a model of the student's rules of physics.

INTRODUCTION

The field of artificial intelligence (AI) has experienced a few innovations which will eventually lead to major advances in the usage of the computer as an educational tool. First of all, the major language of AI, LISP (an anacronym for LISt Programming, is able to be implemented on a microcomputer. Secondly, microcomputers are now powerful enough to support education-related AI software due to faster processing speeds and larger internal memories. The development of intelligent computer-assisted instruction (ICAI) is still at a stage where marketing interests have not corrupted the major goals. It is therefore vitally important to the future of ICAI that a firm foundation of relevant learning theory be incorporated into the prototype systems. Developers of software which is currently in the educational environment seem to place overly-heavy emphasis on the process of teaching, treating it like a simple subroutine. ICAI systems must place emphasis on the process of learning. The word itself is student-centered rather than teacher-centered.

In this paper, we describe the theories of Ausubel's advance organizer,

Gagne's levels of learning, and Bloom's classification system for cognitive behaviors. A paradigm for encoding the expert knowledge of first-semester non-calculus physics into a networked hierarchy (net-archy) of meaningfully-related objectives is presented. Techniques for integration of the resulting physics expert and learning net-archy into the operation of the ICAI system are discussed. The production rules of the expert system are used as the source of expert problem-solving rules.

Learning Theory for the ICAI System

THE ADVANCE ORGANIZER

The many possible methods of transmitting information to a student (ie. discourse, examples, inquiry, etc.) can be summarized by the following models (major theorists in parentheses):

1. Inductive thinking (Taba)
2. Inquiry training (Suchman)
3. Science inquiry (Schwab)
4. Concept attainment (Bruner)
5. Advance Organizer (Ausubel)

The advance organizer model is designed to facilitate information-processing capacities to meaningfully absorb and relate bodies of knowledge (Ausubel, 1963) Its structure and theory, developed before the microcomputer, beg for organization and incorporation within an intelligent CAI system. Ausubel's theory of learning can briefly be described as "a subsumption theory of meaningful learning". The key hypotheses of this theory are:

1. In the learning of potentially meaningful information, meaningful learning is superior to rote learning in terms of amount learned in a given amount of time, retention of what is learned, and the learner's ability to apply what was learned.

2. In the learning of potentially meaningful information, the potential meaningfulness of the material is related to its relative generality or specificity as follows:

The more general the material, the greater the potential meaningfulness.

The more specific the material, the less the potential meaningfulness.

The common approach taken by standard CAI in presenting material to be learned is to follow the sequence of material in a textbook. Unfortunately, textbook authors generally do not adhere to or are aware of Ausubel's hypotheses. New information is normally introduced in such an order that the student can not attach any meaning or significance to it. For example, in his introductory physics textbook, Beuche (1982) introduces SI units, scientific notation, dimensional analysis and vector subtraction in the first chapter. The student is immediately thrust into a position of rote learning since any general subsuming knowledge which may have helped the student to learn the material is at that time unfamiliar. Within the first four pages, the student is informed:

A meter is equal to the length of 1,650,763.73 wavelengths (in a vacuum) of the orange-red line of krypton 86.

The student at this point probably knows nothing of the nature of isotopes, spectroscopy, or the reason for measuring the wavelengths in a vacuum. With this in mind we now turn to an analysis of Ausubel's theory.

Ausubel's (1970) theory is based on the following three concepts:

1. Subsumption: The classification of information within or under a more inclusive category or level of organization. For example, micrometers would be classified under meters; meters under length; length under (among other categories) speed; speed under objects in motion; and objects in motion under kinetics.

2. Meaningful Learning: The learning process through which the learner is presented new material only if it can be related to the learner's present knowledge of a related category. For example, if the learner's present knowledge is:

A scalar quantity has magnitude but not direction.

New meaningful information could be:

Mass is a scalar quantity.

3. Reception Learning: The learning process by which new specific information is presented to the learner in its final form from the tutor. This is the antithesis of discovery learning. For example, the student, after learning the more inclusive category of "gravity on earth", may then be informed that the acceleration due to gravity at sea-level is 9.8 m/s**2.

It must be emphasized that the subsumption theory of meaningful learning is not just designed for presentation of material that is less general than topics already in the student's cognitive structure. This form of information transferral (subordinate subsumption) would be a oneway path toward highly specific information, starting with the student's initial knowledge (see fig. 1). According to the definition of meaningful learning above, new information need only be related to present knowledge. Consequently, there are two other forms of subsumption. The second form of subsumption is super-ordinate subsumption, in which new information presented is related to, but more general than another

category (see fig. 2). As shown in fig.3, the advance organizer facilitates the subsumption of new information at the same level of generality as a relevant category in the learner's cognitive structure (coordinate subsumption). This form assumes the existence of a "parent" cell of information which is related to, but more general than both subsumed categories.

Even though the advance organizer was designed for a discipline that is hierarchical in structure, it can easily be modified to adapt to a system based on a net-archy such as physics. This model assumes that the learner's cognitive structure, or memory organization is also structured according to a net-archy. This is plausible since humans seem to recall facts better when the facts were learned with an associated fact. Most theories of connectionism, dating from Thorndike to the present are unified in the hypothesis that memory is built through connections made with existing memory.

One observation that can be made about this net-archy is that it appears that the model would deny any entry point for anyone unfamiliar with physics. Simply put, there must be some initial knowledge the entering student possesses that can be used by the system as a seed. It must therefore be assumed that the student has some prerequisite knowledge which matches at least one area of the organizer. This assumption will be easily realized in physics. Some examples of prerequisite knowledge the system will expect of the student are that the student is aware of gravity, can judge obvious velocity differences, and has ridden in a car. Besides having a record of all expected areas of prerequisite knowledge, the system will administer an entry-level test to determine other areas of the organizer which can be used as seeds for meaningful learning.

The adaptability of the advance organizer to helping students learn by approaching the goal knowledge from various directions is obviously learner-oriented. Not only can the system execute tutorial strategy rules to control the path of instruction, the system can experiment with variations of the strategy to determine a best overall and best-for-student strategy. The advance organizer should be shown to the student via the personal computer graphics display with the current area of instruction marked to allow the student to be aware of his/her status, of how the student is progressing through the instruction. Feedback from the student at this point can be used to provide the student with a degree of control over the course of instruction. As will be discussed later, the knowledge of a node in the net-archy will be path-dependent. This should be seen as favorable as it will allow lower-performing students to always be advancing, thus reducing the chances of frustration.

TYPES OF LEARNING

Up to this point, the categories within the organizer have not been defined. In order to determine what "bit of physics" needs to be assigned to each category, it is necessary to examine both the structure of the organizer and the student's cognitive structure. At each node, the system will attempt to cause the student to progress toward the educational objective of understanding introductory physics. An understanding of physics requires more than just the assimilation of pieces of information, so the nodes can't all be packed with facts.

Gagne (1974) has identified eight basic types of learning. Assuming that a student already salivates when a pizza is presented, the relevant levels of learning to be incorporated into the physics ICAI system are, from most to least complex:

- Problem solving
- Principles
  - Inferences
  - Communication
  - Space-time relationships
  - Measuring
  - Classification
- Concepts
  - Abstract
  - Concrete
  - Defined
- Multiple discriminations
- Verbal association

Gagne also identifies the need for a cognitive strategy and sets this apart because it is the learner's internal method of handling knowledge. Since a certain strategy can be applied to many problems, cognitive strategy will be included with the learning types between principles and problem solving. Some students may never acquire a cognitive strategy (Piaget would agree), and will need to be cued through or will perform rotely through the solution of a problem.

Application

EXPERT SYSTEM CONSTRUCTION--ORGANIZATION OF KNOWLEDGE

The organizer is constructed in a depth-first manner with each of the nodes being one of the Gagne learning types. The learning types fall naturally into the advance organizer net-archy. The pedagogical expert of physics is

constructed by recursively asking the question "What learning is necessary to accomplish this?" The guiding instructional objective for the system is for the student to be able to solve three basic types of accelerated motion problems.

PROBLEM TYPE 1

Objects with initial horizontal velocity in free-fall.

Several principles are involved, but one is the high-level principle that vertical acceleration is independent of horizontal velocity. This principle, upon recursion of the question "What is required?", is found to require other principles for full understanding. One of these is the classification of horizontal velocity of a projected object under Newton's first law (inertia). This principle will, in turn, require a concept of velocity, which will require a concept of speed. Speed will require the concept of distance which will require a discrimination between measurable lengths. At the level of least complexity, there will need to be a verbal association between the word length and idea of measuring the distance. As shown in Figure 4, the recursion continues by returning to the next-highest level which had a node with an incomplete set of requirements.

Although there will be a limited amount of networking within the learning net-archy for one problem, the networking becomes increasingly complex and networked upon the addition of other problems.

PROBLEM TYPE 2

Object with initial velocity on horizontal surface subjected to net force.

PROBLEM TYPE 3

Object with initial positive vertical component of velocity in free-fall.

These other problems will call for some of the same principles, concepts, and discriminations as previous problems. As each problem type generates its own learning net-archy, the number of new prerequisites decreases due to the more complex network which is formed. This progressive networking is shown in figure 5.

REPRESENTATION OF LEARNING TYPES AS A SEMANTIC NET

Recalling the previous discussion of the different levels of learning according to Gagne (1974), it is apparent that the intelligent system needs to be provided with information as to how, when, why, and in what context nodes relate to other nodes of the net-archy. First of all, the basic terms of physics are entered through the use of identifying the learning types. For example, in order to define "vector" as a concept, the LISP defprop statement is used as follows:

```
(defprop vector concept learning-type)
```

And, of course, others follow, such as:

```
(defprop newtons-first-law principle
     learning-type)

(defprop measuring-lengths discrimination
     learning-type)
```

The structure is connected into a semantic net by putting into LISP notation the relationships developed through the results of the prerequisite recursion. For example, the first plunge into the depth-first recursion of figure 4 yields:

```
(defprop problem-1 vert-acc-indep-of-
     hor-vel needed-principle)
(defprop vert-acc-indep-of-hor-vel hor-
     vel-obeys-newtons-first-law needed-
     principle)
(defprop hor-vel-obeys-newtons-first-
     law velocity needed-concept)
(defprop velocity speed needed-concept)
(defprop speed distance needed-concept)
(defprop distance measuring-lengths
     needed-discrimination)
(defprop measuring-lengths (length-
     thought determine-distance)
     verbal-associations)
```

Another relationship useful in semantic nets is the "isa" relationship:

```
Facts
  ((acceleration isa vector)
   (speed isa scalar)
   (free-fall isa acceleration)
   (dropped-object isa freefall)
   (wind-resistance isa friction))
```

The semantic net is completed through the establishment of the other relationships, such as:

```
(defprop mass (kilograms grams slugs)
     units)
(defprop vector (magnitude direction)
     characteristics)
```

The establishment of this semantic net facilitates the operation of the advance organizer learning strategies. It is through these connections that the flow of instruction may proceed in either an

expert-recommended, student-recommended, or experimental direction. The decision of which entity determines the instructional control is made by a set of high-level rules in the tutorial strategy section of the system.

INCORPORATION OF RULE BASE

The establishment of principles, concepts, and discriminations within the learner's mind, while being a goal of the system, is not sufficient to enable problem-solving ability. The ability of the learner to solve accelerated motion problems involves elements of the cognitive strategy mentioned earlier, the process of envisionment (de Dleer, 1975), and a rule-governed approach (Maloney, 1984). The expert knowledge of the rules of accelerated motion represent the action the expert of physics would take, given certain situations. These rules can number in the hundreds or thousands, due to the complexity or broadness of a subject. The ICAI system GUIDON used the rules of MYCIN (Shortliffe, 1976). Some of the rules of accelerated motion which will be used in this expert system of physics (ESP) include:

```
((rule grav133
    (if (object is dropped)
        (air-friction is zero)
        (init-hor-vel is zero))
    (then (resulting-motion is pos-acc-
    downward)
        (acc-direction is grav-direction)
        (acc-source is gravity)
        (final-hor-vel is zero)))

((rule grav134
     (if (object is dropped)
         (air-friction is zero)
         (init-hor-vel greater-than-zero))
     (then (resulting-motion is pos-acc-
     down-parabola)
         (final-motion-direction1 is
        acc)
         (acc-direction is perp-
        direction1)
         (acc-source is grav)))
```

The expert system now has the following three capabilities:

1. Intelligent problem generation-the conditions in the production rules can guide the formation of problems. For instance, the Ifs in the following rule:

```
((rule inertia51
     (if (initial-motion is const-vel)
         (mag-net-force-on gt zero)
         (friction is zero)
         (net-force-dir is
           perp-to-vel))
         (then (resulting-motion is pos-
acc-perp-parab)
         (resulting-motion-direction1 is
           const-vel)
         (final-vel-direction1 is init-
           vel-direction1)
         (resulting-motion-in-net-force-
           direction is acc)
         (acc-source is net-force)
         (acc-direction is perp-
           direction1)))
```

can be used to construct a problem situation for the student to solve. A problem-generating module will set up the problem, either verbally or graphically using each condition in the If portion of the rule. The system may ask the student to name a moving object, whereupon a successful match with stored figures will cause the desired figure to appear on the screen moving at constant velocity. Various forces will then be shown acting on the object, with their magnitudes and directions given. These forces will be picked so as to yield a net force direction that is perpendicular to the object's initial direction of travel. The learner will be told to ignore frictional effects, and then asked to predict the resulting motion.

2. An explanation facility to allow the system to intelligently communicate with the student. A natural language interface can be used to explain how answers were arrived at, why principles were used, and other knowledge contained within the expert system.

3. A separate set of production rules for what the student believes can be built by setting up the student with the expert's conditions and forming the action statements from the student's response. A pattern match between the student's action and an incorrect rule can then cause the firing of "misconception-buster" remedial instruction path.

PRETEST

Past developments which came about through the use of TURTLE geometry (Ellington, 1981), whereby paths of learning were determined through the use of a game, will be incorporated into the instructional strategy as a pretest game to assist the hypotheses formation section of the software.

MEASURING AMOUNT OF LEARNING

Since the amount of learning at each node is dependent upon factors such as previous misconceptions and path taken, Bloom's taxonomy for assessing cognitive

BEST COPY AVAILABLE

behaviors is used to provide a measure of learning. The classes, from highest to lowest level of behavior are:

6.00 Evaluation
5.00 Synthesis
4.00 Analysis
3.00 Application
2.00 Comprehension
1.00 Knowledge

There are 23 sub-classes within these. Questions can be geared to determine the student's learning level for each node. As shown in Figure 6, the student's level of learning of the concept of force may be determined by gearing the questions to correspond with the behavior taxonomy.

5.00 Synthesis... student is able to modify existing misconceptions of the nature of forces.

4.00 Analysis ... the student checks the consistency of graphical depictions of forces acting on objects.

3.00 Application... the student predicts the probable effects of various forces acting on objects in specified situations.

2.00 Comprehension... the student translates verbal descriptions of applied forces into a mathematical expression.

1.00 Knowledge... the student recalls the definition of force and net force, as well as states the necessary properties of all forces.

Figure 6. Assessing Learning at Each Node The student's understanding of the concept of force can be assessed by designing questions according to Bloom's taxonomy for cognitive behaviors.

The tutorial strategy module of the ICAI system can thus determine the appropriate direction for future instruction. For instance, a 5.00-6.00 performance on a principle may result in the system generating a problem requiring the principle and an untutored principle for solution. This is a favorable situation for the student, as the needed principle then has meaning and will be more readily learned. A 2.00 performance on a principle will result in the system directing instruction to a sub-ordinately subsumed concept which needs to be more thoroughly learned.

CONCLUSION

There are many modules within an ICAI system that must be developed and coordinated to yield a tutoring system which can truly be called intelligent. The student module should not only keep a record of paths taken and learning at each node, it should constantly be hypothesizing the student's current state of knowledge. The tutorial strategy module must decide when the student is confused or bored. Another problem with ICAI via personal computer is memory. The system described here is running on a PC with 640 Kbytes internal RAM and 10 Megabytes on a hard disk. A natural language front-end would take away much of the memory needed for other tasks. This language barrier has been overcome by allowing the student to form questions and comments from a parsible menu. Production rules seem to multiply exponentially as the content is increased. There were hundreds of rules just for simple statics and kinetics problems. It is recommended to the ICAI developers not to work with too much content. Only a seed of content is required to accomplish R&D work. After workable skeletons of the modules are more fully developed, then the complete courses can be encoded. By that time, an intelligent "professor decoder" will have been developed so that the expert can sit at a terminal and emerge the next day having imparted his/her essence to the ICAI system.

BIBLIOGRAPHY

Ausubel, David P. The Psychology of Meaningful Verbal Learning Grue & Stratton, Inc. New York, N.Y., 1963.

Bloom, Benjamin S., Taxonomy of Educational Objectives, Handbook 1, Cognitive Domain David McKay Co., Inc., New York N.Y., 1956.

Clancey, William J., "GUIDON", Journal of Computer-Based Instruction Summer, 1983, 10 #1&2.

de Kleer, J., "Qualitative and Quantitative knowledge in classical Mechanics" Technical Report 352, MIT AL Lab, Cambridge, Mass., 1975.

Ellington, Henry, Addinall, Eric, and Percival, Fred, Games and Simulations in Science Education Nichols Publishing Co., New York, 1981.

Gagne, Robert M. & Briggs, Leslie J., Principles of Instructional Design, Holt, Rinehart, and Winston, Inc., New York, N.Y., 1974.

BEST COPY AVAILABLE

O'Shea, Timothy, Self-Improving Teaching Systems Doctoral Dissertation, 1979.

Shortliffe, E.H., Computer-Based Medical Consultations: MYCIN New York, American Elsevier, 1976.

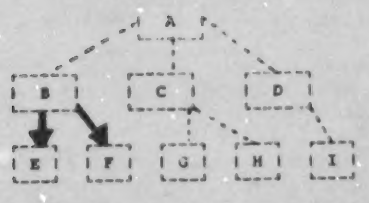


Figure 1. Sub-ordinate subsumption: If current instruction is at B, then instruction may proceed to E or F.

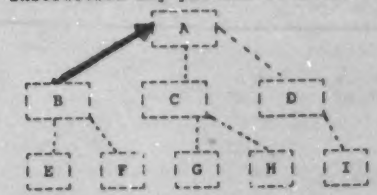


Figure 2. Super-ordinate subsumption: If instruction is at B, then instruction may proceed to A.

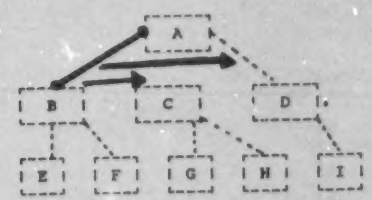


Figure 3. Co-ordinate subsumption: If instruction is at B and a relationship between B and A has been established, then instruction may proceed to C or D.

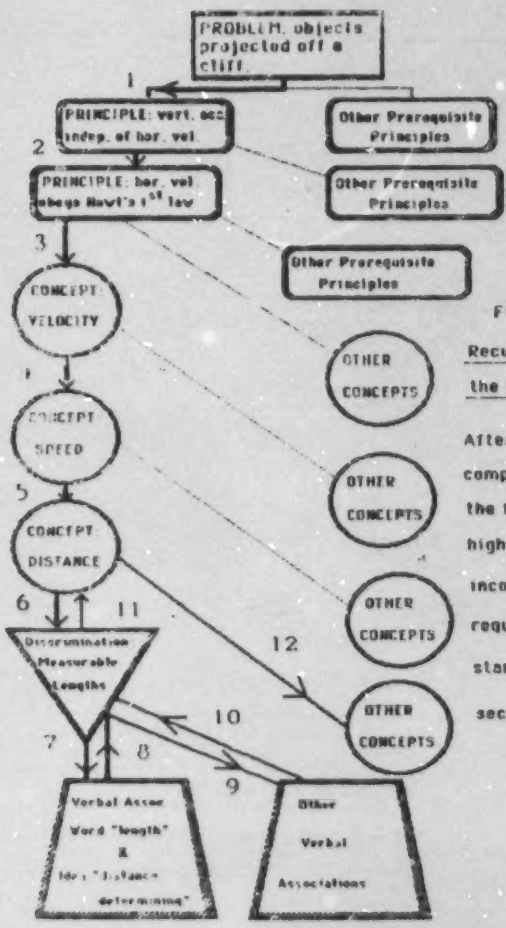


Figure 4:

Recursion for building the physics net-archy

After reaching the least complex learning-type on the first pass, the next-highest node with an incomplete set of requirements is the starting-point for secondary recursion.

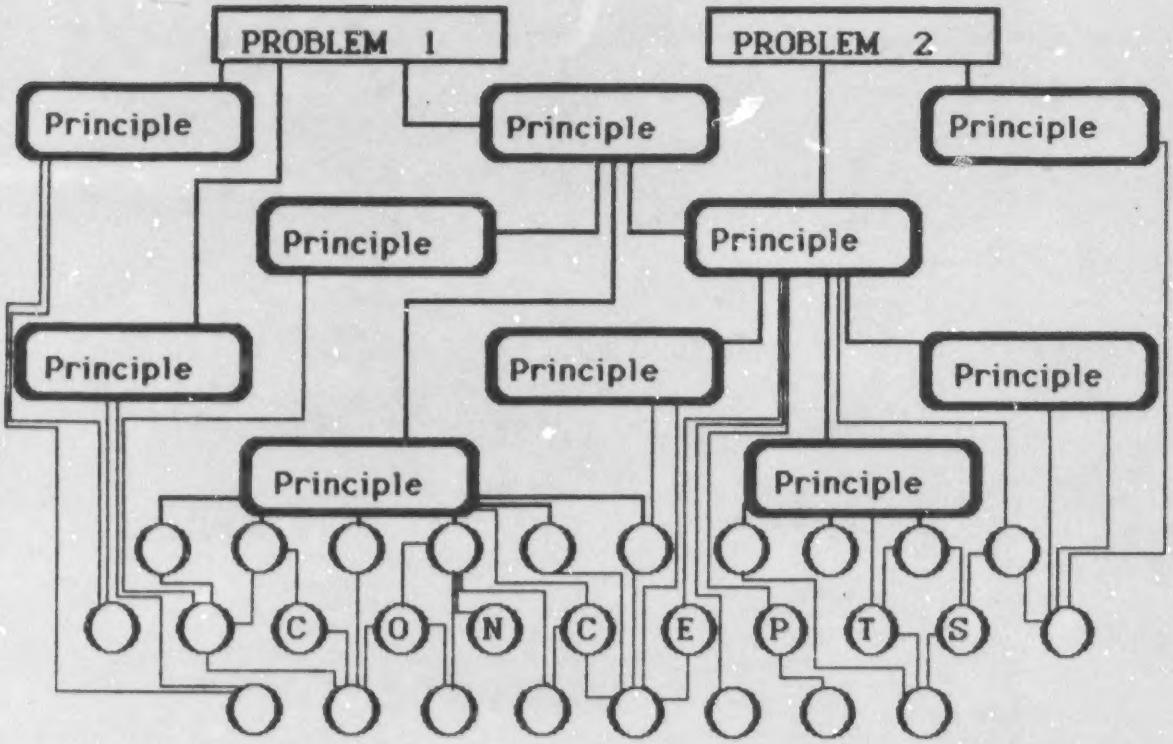


Figure 5: Networking of learning hierarchy (through concept level) for two physics problems related to accelerated motion.

USING COMPUTER GRAPHICS AND ANIMATION IN TESTING

James R. Okey, University of Georgia

Michael E. Hale, University of Georgia

ABSTRACT

The graphics and animation capabilities of computers offer a new dimension in testing. Results from paper and pencil tests may not reflect true ability because of reading comprehension or visualization problems. Computers can present simulated actions and events that are the focus of test questions. The computer animated questions should then be more like learning situations and the real life situations to which learning is applied.

In this study middle school students completed either a paper and pencil or computer presented test of problem solving skills. Reading comprehension and visualization ability scores were also obtained. Analysis of the data showed no difference in test scores for the two groups -- a finding contrary to that in other studies using picture enhanced printed tests.

Relatively low reading ability may have been a major contributor to the lack of an experimental effect in this study. Students had to read the interrogative portion of test questions even though the problem situation was dynamically presented with computer graphics and animation. Familiar problem contexts may also have contributed to the no difference result. Students were aware of the events in the problems even without computer enhancements.

Introduction

The purpose of testing is to gain valid and reliable information. Test results should reflect true and consistent performance of individuals. Test validity and reliability are dealt with by carefully wording items, selecting items of appropriate difficulty, and assuring correspondence of test items and objectives. But despite careful attention to these and other factors, test results may not represent true performance in some situations because of the way tests are presented.

In many higher order areas of learning (e.g., problem solving, application of information, or synthesis of ideas) the person being tested must comprehend both context and actions in order to answer questions correctly. Problem context and actions are typically presented in written form and the burden on the test taker is to convert the printed words into mental pictures of dynamic events. Both poor reading comprehension and lack of visualization ability may handicap test takers to the extent that test scores do not represent true ability.

The graphics and animation capabilities of computers offer a way of addressing some of the problems of testing. Instead of presenting only printed questions or even questions with accompanying pictures, computers can present simulated actions and events that are the focus of test questions. The dynamic capabilities of computers can offer testing situations more in line with the real events students encounter during learning and more like the real life situations in which their knowledge will be applied.

BEST COPY AVAILABLE

Purpose of the Study

The purpose of this study was to examine student performance on printed and computer animated versions of a problem solving test. Further, the intent was to find out if certain student characteristics would influence test performance on the two kinds of tests. The characteristics examined were visualization ability and reading comprehension.

Procedures

A 26-item, problem solving test in science was used in the study (Hale, Shaw, Burns, Okey, 1985) that assessed abilities to identify variables, state hypotheses, define operationally, plan investigations, and graph and interpret data. The written version and the computer animated version of the test asked identical questions. The difference was in how the problems that were the focus of the questions were presented. In the printed test, a written description of a problem situation would be given. In the computer test, computer graphics and animation were used to portray dynamically the phenomena that were at issue in the test questions. For example, the computer animated test showed shadows of different length being cast at different times of the day, projectiles being fired with different charges, basketballs being bounced from different heights, and plants growing under different conditions. In these and other instances, the printed test provided a verbal description of the event while the computer test presented the situation with pictures and animation.

Even though the printed and computer tests were different in how events were portrayed they asked identical questions. Further, the questions printed on the sheets were also printed on the screen. The result was that students using the computer version were assisted by the computer graphics in the presentation of a problem but still had to be able to read the interrogative part.

A total of 94 students in grades 7 and 9 were the subjects in the study. They came from a combined small town/rural school district. Boys and girls were about equally represented in the sample. Approximately 85% of the students were Black with the remainder being Caucasians. Students at each grade level were randomly assigned to either the printed or computer version of the test. Tests were completed in group settings and took about 50 minutes. The computer test was given in a room with 14 Apple computers. Previously students had some but very limited experience with computers.

In addition to completing the test, all students took the Visualizer-Verbalizer Questionnaire (VVQ)--a 15-item scale embedded in a 30-item instrument. The VVQ was developed by Richardson (1977) to measure individual differences in modes of processing cognitive events. The items are designed to find out if a person is able to visualize events (e.g., My dreams are extremely vivid; I cannot generate a mental picture of a friends face when I close my eyes). This instrument was used because it was felt that persons who were poor visualizers might especially be helped by the graphics/animation presentation of test items. Scores on the VVQ were used to divide students into high and low categories.

School records were used to identify reading levels of students. The reading levels were scores from standardized reading tests given by the school during grades 6 and 8.

The data were analyzed using analysis of variance with test type, reading level, grade level, and visualization scores as independent variables and the problem solving test score as the dependent variable. Only 2-way interactions were considered in the analysis after preliminary results showed no interaction effects.

Results

Mean scores on the problem solving test, visualization questionnaire, and reading level measure are given in Table 1. The overall mean score on the problem solving test was only 10.7 (out of 26 items). The mean reading level score in grade equivalent units for the combined grade 7 and grade 9 students was just 6.2. And the mean score on the visualization measure was 7.32 (out of 15 items).

Analysis of variance results (Table 2) failed to show significant differences in scores on the two types of tests. In fact, students completing the written test scored slightly higher. Similarly, significant differences in problem solving test scores between students categorized as low or high visualizers were not found. Significant differences in the scores obtained by students with different reading levels were found. As expected, significant grade level differences were also noted with ninth graders scoring about 12% higher than seventh graders.

Reliability analysis of the tests showed them to be reasonable, if not remarkable, measures of the variables. The reliabilities (Chronbach's alpha) of the computer problem solving, written problem

BEST COPY AVAILABLE

solving, and visualization measures were .66, .71, and .50, respectively. Considering the lengths of the instruments, the age of the students, and the type of material being measured, these are not unusual reliability coefficients.

Conclusions and Implications

Although logical arguments can be made for the value of visual (cf., Dwyer & DeMelo, 1981) and animated computer images during testing, the results of this study do not support this view. Students receiving the computer enhanced test items performed no better than their counterparts who read the test items in a conventional way.

Further logic, however, suggests that the issue of the value of computers in simulating events during testing should not die. This study provided one look at the problem but surely not a definitive one. Factors which might have affected the outcome are numerous.

1. The students are, in the main, poor readers. On average they read about two years below grade level. In both versions of the test students still had to read and respond to printed test questions.
2. The students taking the computer version of the test had little computing experience. Observations made during the test were that they needed to focus a great deal on the computer and its operation perhaps detracting from attention to the test itself.
3. The problem solving tasks chosen for testing are so familiar that they may not require much visual assistance through computer animation. If students already have good mental images of basketballs bouncing, planes flying, or plants being watered and fertilized, the computer images may add little to what a student already knows.
4. The visualization questionnaire may have been beyond the reading capability of these students. The mean score on it was about at the chance level.
5. The computer screen displays may not have been as clear to students as they were to adults. Student comments suggested that the images were sometimes misunderstood.

Additional work needs to be pursued in the study of computer enhanced testing. Students with better reading ability need to be studied. Problem selection for the computer animation needs to be altered so that prior familiarity with the tested ideas can be discounted. Screen displays need to be developed with attention to clarity and visual contrast (cf., Alexandrini, 1984).

REFERENCES

Alesandrini, K. (1984). Five star screen design. Data Training, 39-41.

Dwyer, F. & DeMelo, H. (1981). The effect of visual testing in assessing the instructional potential of variables associated with visualized instruction. Journal of Instructional Psychology, 10(3), 126-141.

Hale, M., Shaw, E., Burns, J. & Okey, J. (1985). Using computer animation in science testing. Computers in the Schools, 2, (in press).

Richardson, A. (1977). Verbalizer-visualizer: A cognitive style dimension. Journal of Mental Imagery, 1, 109-126.

TABLE 1

Descriptive Data

Measure	Problem Solving Test: Printed	Computer	Combined
Problem Solving Test (26 items)	11.88	10.28	10.69
Visualizer-Verbalizer Questionnaire (15 items)	7.31	7.38	6.19
Reading Level (in grade equivalents)	5.93	6.43	7.32

TABLE 2

Analysis of Variance Results for Two Types of Tests *

Source	df	SS	F	p
Test Type	1	15.85	1.53	.22
Visualization Ability	1	0.12	0.01	.91
Reading Level	2	586.84	28.38	.00
Grade	1	82.10	7.94	.00
Error	65	672.06		

*Interaction terms were not significant and were left out of the table.

250

ON CONCEPTUALIZING RESEARCH METHODOLOGIES FOR COMPUTER-BASED INSTRUCTION

DAVID REINKING

RUTGERS UNIVERSITY

ABSTRACT

Current research methodologies may be inadequate to significantly enhance our understanding of computer-based instruction (CBI). Inconsistent findings and methodological shortcomings in media research, including CBI, raise this possibility. This paper suggests that considering fundamental issues in educational research may help researchers conceptualize more productive research methodologies for CBI. An examination of these issues should lead to specific recommendations for conceptualizing and conducting experiments involving the computer in instruction. Specific recommendations are discussed in the following categories: moving towards theory building, expanding research methodologies, and setting priorities for applied research.

The pitfalls of educational research are legion. The risks of succumbing to these pitfalls may be amplified when research involves innovative technologies like the computer. The exciting potential of the computer for enhancing instruction generates spontaneous enthusiasm and a concomitant need to justify this enthusiasm. Consequently, there may be a temptation to short-circuit the research process by asking superficial questions, by sacrificing sound methodology and by over-generalizing findings. The history of research investigating a variety of instructional media also suggests caution for those interested in researching topics in computer-based instruction (CBI). Several writers have chronicled the failure of researchers to create a useful research base for guiding the selection and use of instructional media (Clark and Bovy, 1983; DiVesta, 1975; Jamison, Suppes, and Wells, 1974; Leifer, 1976; Oettinger and Zapol, 1971; Saettler, 1968).

Existing CBI research appears to fare no better. Evidence that methodology, for example, is a serious concern can be found in Kulik's (1983) meta-analysis of CBI studies in which a majority of the available studies were eliminated due to methodological shortcomings. Before we accept the possibility that strong empirical evidence is not possible or alternatively continue to muddy the empirical waters, we should compare the goodness of fit between research methodology and the types of questions for which we seek answers. The purpose of this paper is to highlight a set of conceptual and methodological considerations which may be relevant to such a comparison.

FUNDAMENTAL CONCEPTUAL AND METHODOLOGICAL CONCERNS

Research methodology is presumably a function of the types of questions the researcher chooses to address. A conceptualization of a research study begins, therefore, with a question in the mind of the researcher. The old saw about getting a good answer only if there is a good question is important at this stage but the process is more complex when put into the context of educational research. The researcher must also consider potential answers to the question, how those answers might be explained, and use these notions to formulate a methodology for research. The challenge of research is not simply generating good questions, but evolving strategies for generating a limited set of answers.

Put another way, the fundamental issue is whether or not research methodology will permit what Platt (1968) has termed strong inferences as opposed to weak generalizations. Strong inferences are, of course, preferred but are similarly more difficult to ferret out. They demand careful control of variables which sometimes means a movement towards "laboratory" as opposed to "real world" conditions. Achieving strong inferences is also facilitated by the presence of a guiding theory. A theory enables experimental results to be interpreted as "a case in point" as opposed to an isolated phenomenon with many alternative explanations. Without a theoretical perspective isolating a set of significant variables for study becomes difficult.

As an example consider the following question which is typical of those motivating existing CBI research: Can a computer help students learn concepts in biology? Without theoretical guidance or a concern for the level at which findings might be generalized a conceptually simple experiment may emerge from this question. A biological concept is selected and taught via a computer to an appropriate population after which some achievement measure is employed to compare these subjects to others taught the same concept via alternative media. The best a researcher can do using this methodology is to report results and speculate broadly as to what may have caused them. At worst, a misguided or over-enthusiastic researcher will on occasion use the resulting data to make general statements concerning the usefulness of computers to teach biological concepts.

Raw empiricism is the dominant characteristic of this type of study and many of the benefits of experimental research are lost or risk being perverted. Very little is contributed to an overall understanding of the computer in instruction or for that matter its utility in teaching biological concepts beyond the specific conditions of the experiment. Methodology in this case has been conceived as only a logical extension of the question without considering the realm of possible answers. Current conceptualizations of CBI research methodology may suffer from this fundamental weakness.

SPECIFIC CONCEPTUAL AND METHODOLOGICAL RECOMMENDATIONS

Even a casual review of CBI research indicates a tendency to duplicate the methodologies of other media research with little reason to believe that results will be any more enlightening. One way to address this concern is to re-examine the conceptual and methodological foundations of research involving instructional media.

To be valuable, however, this exercise must result in specific recommendations which can guide the researcher. Below is an attempt to move in this direction. Recommendations have been grouped into three broad categories: moving towards theory building, expanding research methodologies, and setting priorities for applied research.

Moving Towards Theory Building. Lachenmeyer (1970) has argued that the predominant view of experimentation in the social sciences has been inefficient in that it does not facilitate the formulation of general theories. The typical research study in psychology and education investigates isolated hypotheses which are generated primarily from a review of previous research as opposed to observable phenomena. Presumably, the cumulative effect of many experiments will be the development of a general theory which consolidates findings into a unified whole. This has rarely, if ever, occurred, however, in educational research.

Most successful theory building, on the other hand, occurs when general principles are inducted from directly observable, naturally occurring phenomena. The initial goal of the researcher is to develop adequate measurement instruments and methodologies for studying a readily observable phenomenon. Later, after fact finding pilot studies, a theoretical statement may emerge to explain facts related to the phenomena studied. At this point, the theoretical statement can be used deductively to generate hypotheses for experimental verification.

The need for developing a theoretical orientation to media research has been articulated by a number of writers. Salomon (1979) and Salomon and Clark (1977), for example, have attributed the lack of consistent findings in media research to the absence of a theoretical orientation. They feel that most of the existing research is a result, in their words, of investigations with media to determine instructional effectiveness. Instead, they propose that a theoretical orientation leads to research on media to determine psychological effects. This puts the investigation of instructional media like the computer within the realm of psychological and educational theory instead of existing in a theoretical vacuum. Similarly, Clark and Bovy (1983) have cautioned against the confounding of technology and instructional method in formulating and conducting research involving instructional media. Ellis (1976) presaged these positions when he argued that understanding the use of computers in education really begins with an understanding of education.

A move towards a theory building orientation in CBI research would have significant impact on research methodologies. More studies would be conceived, carried out, and interpreted as pilot or exploratory studies. The importance of studies which focus on sharpening measurement instruments and developing workable methodologies would be recognized. In addition, the likelihood that findings would be over-generalized would be reduced. In short, even the most applied research would be judged in terms of its contribution to a broader understanding of the computer's role as a medium of instruction.

At the same time focusing on theory building would provide a rationale for basic research in CBI. Basic research, for example, is possible when the computer is integrated into a more general theory of instructional media. Theoretical positions like those developed by Salomon (1979) and Olson (1976) could be used to guide the conceptualization of basic research hypotheses and the development of suitable methodologies. By carrying out basic research founded on accepted theoretical positions researchers would be forced to confront and tease out the significant attributes of the computer as a medium of instruction. We would see fewer studies which make recommendations as to whether the computer is a viable medium for instruction and perhaps more that would give us insight into when, where, and with whom the computer might best be suited for particular categories of instructional content.

Expanding Research Methodologies. Building theory implies a search for those variables which may account for or predict the occurrence of a range of phenomena. When dealing with human behavior and learning, the researcher is assured that the number of variables and their interactions will be enormously complex. While a theory will reduce the number of variables to consider relevant for study, considerable complexity remains. When investigating instructional uses of the computer, identifying relevant variables becomes even more difficult. Computer technology makes the options for delivering instruction so open ended that even the set of possible variables may not be intuitively obvious (Reinking, 1984).

What does this imply specifically for research involving computers in instruction? In general this means expanding methodology to recognize a wider range of variables and to permit those which are significant to surface.

Methodologies which recognize and pursue only technological differences may be too simplistic. Few useful inferences can be made about the computer's relationship to instruction when so many potentially significant variables are ignored.

A desire to contend with more variables may dictate the need for more complex methodologies in experimental research. A wider range of statistical techniques may be necessary; multivariate and regression analyses may need to supplement univariate ANOVAs. Even simple designs and straightforward statistical analyses, however, can be made more powerful by including multiple dependent measures and attempting to replicate results. The latter option will, of course, require the cooperation of journal editors. The experimental bias of educational and psychological research may also need to be tempered in order to encourage non-experimental methods employing correlational analyses or even case studies. Caution may also be necessary in foreclosing areas of inquiry on the basis of accepting a single null hypothesis.

Examples of how creative research methodologies could proceed from theoretical underpinnings and a concern for relevant variables can be found in Salomon and Clark's (1977) examination of research methodologies for media research. They have offered several viable research designs and statistical techniques which would be directly applicable to CBI research.

More productive methodologies may also be a result of collaborative research which bring together colleagues of varying expertise. The interacting variables which operate simultaneously when using the computer in instruction can rarely be seen clearly by one individual. Ideally expertise would be sought out to insure knowledge of media, the instructional content and its pedagogy, and cognitive psychology. Even these areas do not exhaust all of the sources of significant variation (eg. social and environmental factors).

Setting Priorities for Applied Research. A perhaps healthy tension has always existed between basic and applied research. Under ideal circumstances these research emphases complement each other so that both theory and practice are mutually enriched. Under less than ideal circumstances the distinction between the two is unclear and research proceeds haphazardly with little clear direction. One of the theses of this paper is that the latter condition is more characteristic of media research in general and CBI research in particular.

A common misconception, however, is that basic research is uniformly more essential to the theory building process. Lachenmeyer (1970) has cited several examples from the history of science which suggests that the opposite is more likely the case. Theory building is at first more often a result of observing events in the real world which is the domain of the applied researcher. The applied researcher, however, must recognize the unique contribution applied research can make to theory building and conceptualize methodologies accordingly.

This means more than suggesting theoretical explanations for findings; it also means seeking out real instructional problems that may contribute to a broader theoretical understanding of CBI. Wilkinson (1984) has suggested priorities for selecting research topics to explore computer applications in the teaching of a content area. First, the computer should be used to manipulate content in a fashion which is not readily duplicated by other instructional media. Secondly, the use of the computer should be guided by accepted pedagogical principles for teaching the content selected. Finally, a high priority should be given to research which addresses areas of instruction which have proven to be problematic.

Researchers who subscribe to these priorities will increase the likelihood that their research will be more relevant to the practitioner as well as contributing to the theory building process.

CONCLUSION

In summary, current research methodologies may be inadequate to significantly enhance our understanding of CBI. Inconsistent findings in media research, including CBI, raise this possibility. Considering fundamental issues in educational research suggests a broader range of research methodologies may be appropriate. The goal of CBI research should probably be identifying significant variables in an effort to build theoretical perspectives. Finally, the topics of applied research may need to be evaluated closely to insure their usefulness for both practice and theory development.

References

Clark, R. E. and Bovy, R. C. "A Cognitive, Prescriptive, Theory of Instructional Method." mimeographed paper, University of Southern California, 1981.

DiVesta, F. J. "Trait-treatment Interactions, Cognitive Processes, and Research on Communication Media." AV Communication Review, 1975, 23, 185-196.

Jamison, D., Suppes, P. and Wells, S. "The Effectiveness of Alternative Instructional Media: A Survey." Review of Educational Research, 1974, 44, 1-68.

Kulik, J. A. "Effects of Computer-Based Teaching on Secondary School Students." Journal of Educational Psychology, 1983, 75, 19-26.

Lachenmeyer, C. W. "Experimentation--A Misunderstood Methodology in Psychological and Social-Psychological Research." American Psychologist, 1970, 25, 617-624.

Oettinger, A. G. and Zapol, N. Will Technology Help Learning? Report prepared for the Carnegie Commission on Higher Education, December, 1971.

Olson, D. R. "Towards a Theory of Instructional Means." Educational Psychologist, 1976.

Platt, J. R. "Strong Inference." Science, 1964, 146, 347-352.

Reinking, D. P. "Reading, Computers, and a New Technology of Print." in Blanchard, J. S. and Chang, F. (eds.), Computers and Cognitive Processes in Reading. in press.

Saettler, P. "Design and Selection Factors." Review of Educational Research, 1968, 38, 115-128.

Salomon, G. Interaction of Media, Cognition, and Learning. San Francisco: Jossey-Bass, 1979.

Salomon, G. and Clark, R. E. "Reexamining the Methodology of Research on Media and Technology in Education." Review of Educational Research, 1977, 47, 99-120.

Wilkinson, A. C. "Reading in Real Time." in Wilkinson, A. C. (ed.), Classroom Computers and Cognitive Science. New York: Academic Press, 1984.

COMPUTER-MEDIATED SMALL-GROUP DISCUSSION AND ITS POTENTIAL FOR DATA COLLECTION

A.J.L. SINCLAIR

UNIVERSITY OF THE WESTERN CAPE

ABSTRACT

The systematic description of cognitive processes based on data from verbal reports has a long history. The most generally accepted methods of describing cognition consist of thinking aloud, introspection, and protocol analysis. The paper considers the inherent weaknesses of these techniques. The rationale behind computer-mediated small-group discussion as a learning arrangement is given, and the practical application of such arrangement discussed at the hand of an outreach programme to black and coloured final-year high-school pupils. In order to probe deeper into the pupils' problem-solving reasoning, the groups were audiotaped unobtrusively. The tentative results indicate that the taped information will be a valuable source of data on the problem-solving process.

1. Introduction

Present-day techniques for data-collection on problem solving show inherent weaknesses in that they are either pre-planned exercises and therefore to a large extent superficial, or mere reflections on what was supposed to happen during the process.

A much more reliable technique to find out what happens during problem solving may be to audiotape the problem-solving discussions of small groups of learners in front of a terminal. The rationale behind and practical implementation of such a computer-based small-group learning arrangement - and its potential for data collection - is discussed in this paper.

2. Existing data-gathering techniques

The systematic description of cognitive processes based on data from verbal reports dates back to the Wurzburg School psychologists at the turn of the century (Boring, 1950). Introspection and protocol analyses have since been extensively used as methods of describing the learning process. These learner accounts - either recorded as a form of diary or obtained through interviews - of their self-perceptions, self-discoveries and self-revelations are still regarded as the most direct source of insight into the largely unobservable processes of learning (Kidd, 1976; Harri-Augstein, 1977; Ericsson and Simon, 1980).

Cohen and Hosenfeld (1981) designed a three-dimensional model for researching mental states. In thinking aloud the learner lets his thoughts flow verbally without trying to observe them. These verbalisations are concrete, descriptive and unanalysed. Self-observation implies that the learner inspects his mental state, and the resulting verbalisations vary from those that are largely unanalysed and concrete to those that are analysed more extensively and are predominantly abstract.

The difference between the two methods is the time factor. Whereas in "thinking-aloud" data there is invariably only a few seconds' delay between the occurrence of the thought and its verbalisation, self-observation may take place within a few seconds of the thought process - immediate retrospection - or only after a more substantial time-lapse - delayed retrospection. The length of time between process and report can be a serious problem because of the strain on memory to reconstruct the route by which the solution to a problem was arrived at (Smith and Miller, 1978; White, 1980).

The content, the third component of the model, focusses on the academic task at hand. The learner must first be trained to think aloud or retrospect on what he does as he progresses.

The relative merits of these techniques have been argued thoroughly (Nisbett and Wilson, 1977; Smith and Miller, 1978; Ericsson and Simon, 1980; White, 1980). There are, however, inherent weaknesses in all of them. Since the learner has to be trained to think aloud, it cannot be assumed that the problem-solving process is unaffected. Furthermore, the extent to which the thought process can be verbalised remains an open question. Linguistically speaking, verbalisation is by its very nature a filtered end-product.

In the case of retrospection, it has been posited that the learner does not do so on the basis of true introspection, but by applying or generating causal theories about the effects of that type of response. In other words the learner makes a judgement about the probability that a particular stimulus has influenced a certain response. But stimuli are embedded in complex thought configurations and it is highly unlikely that direct inferences - cause is equal to effect - can be made accurately (Nisbett and Wilson, 1977).

It seems therefore that what we are looking for is a learning situation where one has a "running mental model" (de Kleer and Brown, 1981) of problem solving. A small-group learning situation seems to present a possible solution.

Looking critically at existing data-collection techniques, it is as if researchers have forgotten that what we are in effect investigating is how the process of problem solving is executed. Whether this execution is done by the individual learner or in a group situation is of secondary importance. Instead, researchers have fixed their attention on the individual learner as the only source of information.

3. Small-group learning

During the past decade a large number of studies have investigated the relative effectiveness of small-group, large-group and individual learning methods. The positive outcomes of cooperative small-group learning have been found across all disciplines and among students of all ages: greater enjoyment, involvement, motivation and achievement, improvement in understanding of others' perspectives, and increased self-esteem (Johnson and Johnson, 1975, 1978; Slavin, 1980(a), 1980(b); Sharan, 1980; Johnson, 1981).

Student interaction during small-group work has been directly related to the outcomes of the learning process (Webb, 1977). Students who actively taught group members or who received explanations from group members performed better on an achievement test after small-group learning than students of comparable ability who did not participate (Johnson et al., 1981).

These findings corroborate aptitude treatment interaction studies which have suggested that low ability students benefit more by learning in small mixed ability groups rather than in large groups or individually (Webb, 1977; Peterson et al., 1981). This is so since high ability students most often explain in heterogeneous ability groups while low ability students are most often the target of the explanations (Webb, 1980).

Gall and Gall (1976) and Slavin (1977) point out that group feedback and sharing of resources help group members reshape their

ideas and learn new information that they might not discover on their own. The individual learner plays an active role in restructuring this new information (Wittrock, 1974). In other words, the learner generates associations between new information and concepts already learned - a key premise in learning theory.

Although the benefits of small-group learning are ascribed to interaction and communication within the group, thereby enhancing cognitive learning, few studies have systematically observed this interaction. A major difficulty is the lack of reliable observation procedures. Audiotaping and videotaping have been suggested as means of caputuring the rich detail in group discussion.

4. Computer-based instruction (CBI) and the learning of mathematics

The potential of the computer as an instructional medium has been widely commented on. It is especially in the management of learning that its full instructional potential comes to fruition. Computer-managed learning implies individualised instruction that is based on carefully defined objectives, is hierarchical in its content, modular in its presentation and assessment, with diagnostic achievement tests and immediate feedback on student progress built into the system. In this mode the computer is programmed to structure, segment and direct learning to suit the individual learner.

A heuristic approach to the teaching of mathematics has been advocated by many mathematicians and mathematics educators (Libeskind, 1977). Avital (1973) emphasizes the gain in comprehension and retention when the teaching is done heuristically. He states as follows:

> In exposing a student to logically structured, sequentially ordered material, we can improve the learner's grasp of it if we succeed in arousing his awareness of the need for each step.
>
> In efficient expository teaching of mathematical proof, we have to present the material step by step so that the learner first sees the need for the next step, then becomes aware of how each step provides the next. Only if care is taken to organize and present the material in this way can we hope that the learner will be able to store it and retrieve it when necessary. It should be clear that the greatest facilitation is obtained when a rationale can be given for each step that has to be taken. We shall call such a procedure a naturally meaningful one.

The resemblance between the heuristics underlying computer-managed learning and the heuristics for the teaching of mathematics will be apparent.

5. Small-group learning - the UWC solution

The University of the Western Cape (UWC), a community-centred institution in the Republic of South Africa, three years ago initiated a computer-based outreach programme in mathematics and physical science for black and coloured school leavers (Sinclair, 1984).

The goal of the programme is twofold, viz. in the short term to improve the pupils' achievement in the subjects concerned, and in the long run to enhance their learning efficiency, i.e. their thinking skills.

The pupils come predominantly from low-skill, blue-collar and in many cases unstable households. A large part of the black pupils, for example, come from squatter areas. The child in such a household does not get mediated learning experience from the parent, with the result that his thinking skills do not develop as they should. Such a learning-disadvantaged child shows a level of cognitive activity which does not reflect his true cognitive potential. There are also other characteristics associated with a deprived milieu: elevated levels of anxiety, low levels of curiosity and exploratory behaviour, a characteristic orientation towards failure-avoiding rather than towards success-striving, task-extrinsic rather that task-instrinsic motivational orientation, and a negative self-concept.

It is our firm belief that the creation of a rich learning environment can overcome these deficiencies. Such an environment requires that both the content and the mode of instruction be such that they will stimulate the child's thought processes.

The sociocentric upbringing of the child in a working-class family - large groups of children left all day in the care of the family members - and the communal or group orientation of the black child in particular, seem to suggest that a cooperative, non-threatening learning situation, especially at the initial stage of remediation, is essential for the development of the confidence needed to excel as an individual.

A computer-based small-group learning arrangement presents itself as a particularly appropriate solution. We place three to four pupils per terminal. The subdivision into subgroups is based primarily on personality criteria and established friendships rather than on ability. The division is therefore basically a natural one.

We make use of the PLATO Learning Management software package (PLM) which has the following features: testing, diagnosis, prescription of learning resources, retesting, data-collection.

The lesson material which is used consists at base level of instructional units (IU's), each with its own heuristic paradigm. The ability of the student is tested by banks of questions, and depending on the outcome he proceeds or has to work through prescribed material again, which is either on-line or available in print.

6. Computer-mediated small-group discussion as a source of data

The PLM data-collection facility enables the teacher to monitor the progress of the class as a whole or of the individual small groups, their course progress, average module test duration, average number of tests in a particular unit, the number of tests taken before mastery, etc. But

this information, however valuable it may be, does not give the teacher any information on why some groups fall behind or why the class as a whole has difficulty with certain sections of the work. All that it actually does, is to provide statistics on learner performance, i.e. on the product, not the process.

Close observation of the small groups in front of the terminals over a period of time gave us the idea to audiotape their discussions unobtrusively as they proceeded through mathematical problem solving. This venture is still in its infancy and an in-depth analysis of the material will take a long time. The first few tapes, however, show exciting possibilities of penetrating the learners' reasoning.

A few preliminary remarks:

1. There is a remarkably smooth flow of discussion. The pupils constantly check and balance one another. They probe possibilities, offer solutions, and seek confirmation from each other.
2. Interaction with the terminal brings about task-oriented behaviour and gives direction to the discussion.
3. There is clear evidence that the groups want to solve the problems without making use of the HELP key, which is a positive sign. On the other hand, unnecessary time can be wasted this way. The HELP execution paradigm is a learning heuristic in itself: try again ⟶ ask for a clue or first step in that particular problem-solving technique ⟶ ask for a more detailed clue or the next step in the problem-solving technique ⟶ ask for the full solution, correctly set out as a "model answer".
4. The groups find it difficult to relate the verbal formulation of the problem to the underlying mathematical relationships. The black students in particular have problems with "relation" terms.
5. Our learners seemingly lack a task-specific frame of reference. What they apparently need is "schema" knowledge which will enable them to know what information to look for and how to deal with it; in other words they require task-specific rather than problem-specific knowledge.

Estes argued as long ago as 1978 that sound problem-solving theories would result from the recording of factual accounts of what problem solvers do and from inferred concepts of their covert behaviour. The data collection of computer-mediated small-group discussion may be a small step in that direction.

REFERENCES

AVITAL, Shmuel M. Teaching a Mathematical Proof. Int. J. Math. Educ. Sci. Technol., 1973, 4, 143

BORING, E.G. A History of Experimental Psychology, (2nd. ed.). New York: Appleton-Century-Crofts, 1950.

COHEN, A. and HOSENFELD, C. Some Uses of Mentalistic Data in Second Language Research. Language Learning, 1981. 31, 285-313.

DE KLEER, J. and BROWN, J.S. Mental Models of Physical Mechanisms and Their Acquisition. In J.R. Anderson (ed.) Cognitive Skills and Their Acquisition. Hillsdale, N.J.: Lawrence Erlbaum Ass. Inc., 1981.

ERICSSON, K.A. and SIMON, H.A. Verbal Reports as Data. Psychological Review, 1980, 87, 215-251.

ESTES, W.K. The Information-processing Approach to Cognition: A Confluence of Metaphors and Methods. In W.K. Estes (ed.) Handbook of learning and Cognitive Processes: Vol. 5, Human Information Processing. Hillsdale, N.J.: Lawrence Erlbaum Ass. Inc. 1978.

GALL, M.D. and GALL, J.P. The Dimension Method. In N.L. Gage (ed.) The Psychology of Teaching Methods. Chicago: The University of Chicago Press. 1976.

HARRI-AUGSTEIN, E.S. and THOMAS, L. Learning how to Learn. In M. Howe (ed.) Adult Learning. New York: Wiley and sons, 1977.

JOHNSON, D.W. and JOHNSON, R.J. Learning Together and Alone: Cooperation, Competition and Individualization. Englewood Cliffs, N.J.: Prentice hall, 1975.

JOHNSON, D.W. and JOHNSON, R.J. The Instructional Use of Cooperative, Competitive and Individualistic Goal Structures. In H. Walberg (ed.) Educational Environments and Effects. Berkeley, Calif.: Mc Cutchan, 1978.

JOHNSON, D.W. Student-student Interaction: The Neglected variable in Education. Educational Researcher, 1981, 10, 5-10.

JOHNSON, D.W., MARYAMA, G., JOHNSON, R., NELSON, D. and SKON, L. Effects of Cooperative, Competitive, and Individualistic Goal Structures on Achievement: A Meta-analysis. Psychological Bulletin 1981, 89, 47-62.

KIDD, J.R. Adult Learning in the 70's. In R. Smith (ed.) Adult Learning: Issues and Innovations. De Kalb: Northern Illinois University, 1976.

LIBESKIND, S. A Problem-solving Approach to Teaching Mathematics. Educational Studies in Mathematics, 1977, 8, 167.

NISBETT, R.E. and WILSON, T.D. Telling More Than We Can Know: Verbal Report on Mental Processes. Psychological Review, 1977, 84, 231-259.

PETERSON, P.L., JANICKI, T.C. and SWING, S.R. Ability x Treatment Interaction Effects on Children's Learning in Large-Group and Small-group Approaches. American Educational Research Journal, 1977, 18, 453-473.

SINCLAIR, A.J.L. Computer-based Math and Science Compensatory Instruction for School Leavers - A Case Study. 10th International Conference on Improving University Teaching, Maryland, 1984.

SLAVIN, R.E. Classroom Reward Structure: An Analytical and Practical Review. Review of

Educational Research 1977,. 47, 633-650.
SLAVIN, R.E. Cooperative Learning. Review of Educational Research, 1980, 50, 315-342.(a)
SLAVIN, R.E. Cooperative Learning in Teams: State of the Art. Educational Psychologist, 1980, 15, 93-111.(b).
SMITH, E.R. and MILLER, F.S. Limits on Perception of Cognitive Processes: A Reply to Nisbett and Wilson. Psychological Review 1978, 85, 355-362.
WEBB, N.M. Learning in Individual and Small Group Settings (Tech. Report no. 7) Stanford, Calif.: Aptitude Research Project, School of Education, Stanford University, 1977.
WEBB, N.M. A Process-outcome analysis of Learning in Group and Individual Settings. Educational Psychologist, 1980, 15, 69-83.
WHITE, P. Limitations on Verbal Reports of Internal Events: A Refutation of Nisbett and Wilson and of Bem. Psychological Review, 1980, 87, 105-112.
WITTROCK, M.C. Learning as a generative process. Educational Psychologist, 1974, 11, 87-95.

RESEARCH ON USER INTERFACE DESIGN

AS RELATED TO COMPUTER-BASED INSTRUCTION

Mary L. Stoddard - Los Alamos National Laboratory*

ABSTRACT

The fields of interactive software user interface design and computer-based instruction clearly overlap. Work in computer-based instruction (CBI) in such areas as screen design, answer judging, and accommodation of variable user skill levels can contribute to the interface design field. In turn, we in the CBI field could benefit from examining the abundant literature on user interface design, particularly in the areas of user modeling, interface guidelines, and user interface management systems. In this paper, user interface study is defined and some principles of interface design are presented. In addition, specifics are given on how the two fields interrelate.

I. INTRODUCTION

The field of user interface design has emerged in the past few years to become the focus of much work in computing. The reasons for this change are numerous. They include, increase in usage of computers by professionals not skilled in their use, user demand for "friendly" systems, recognized need to reduce user costs, and improved software and hardware. Central to the design of effective human-computer interfaces is one's knowledge of the user. Research studies in the field focus not only on concepts of software design and evaluation but also on building our knowledge base of people's behavior with computer systems. This new field is drawing professionals from cognitive psychology, linguistics, documentation, and human factors, as well as from many areas of computer science.

The relationship between the fields of computer-based instruction (CBI) and user interface design is not well established. This is ironic in that CBI professionals writing courseware have been concerned with effective user interface design all along and thus have something to contribute to the effort. In turn, advances in design methodologies for user interfaces can contribute to current CBI research and development efforts. The purpose of this paper is to provide CBI professionals with some background about the field of user interface design and to discuss how the fields can logically relate. In addition, many key papers in the field are cited here that provide a good starting point for those wishing to get further into the literature. This information should be of particular interest to those designing or evaluating authoring systems and courseware.

II. THE SCOPE OF USER INTERFACE DESIGN

The user interface can be defined as the boundary between the computer and the user. The interface includes dialogues, screen displays, and input devices. Don Norman (1983) suggested a model for interface design wherein the user has a mental model of the process, the designer has a conceptual image of the system, and the physical device has a system image. A key for us as designers is understanding what the user's mental model is.

In this paper, only the software aspects of user interface design will be considered. Hardware aspects of interface design are considered part of the field of ergonomics. Human Factors Society publications are good sources of information in this area.

---

*This work performed under the auspices of the U. S. Department of Energy.

The user interface is only one factor contributing to the behavior of the computer user. Other factors include a user's goal, knowledge, and processing limits (Moran, 1981) (see Figure 1).

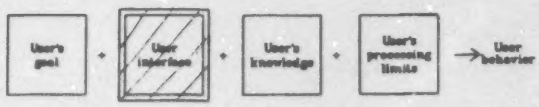


Figure 1. Factors contributing to the behavior of the computer user (Moran, 1981).

III. SOME PRINCIPLES OF USER INTERFACE DESIGN

There have been a multitude of papers published about user interface design, yet no one textbook contains a concise overview. As a result, it is difficult for a person not deeply involved in the field to quickly obtain a summary of what has been learned. The field is young and this phenomenon is characteristic of any new science. In this section, an attempt is made to provide a brief survey of some general design principles. Be aware that, as there is no one "theory of user interface design," there is no consensus in the field, and the word principles should be loosely interpreted. In addition, the following discussion is by no means exhaustive of the subject. It serves at a high level; Smith and Aucella (1983), among others, have published detailed guidelines for user interface design.

Early knowledge of users. Users may not always be "right" about what are effective interface characteristics of a system, but to attain the goals of ease-of-use and ease-of-learning, one must understand who the users will be and how they will be using the system (Gould and Lewis, 1983).

Early testing with a prototype system. Often in the development process testing with users is not done until the software is up and running. Such testing is often done just to get out the bugs. If, however, testing is done at an early design stage and with simulations (e.g., on paper, through video, and through mock-up of software), many factors that designers cannot guess accurately can be detected (see Figure 2). Examples include user reaction to particular menus or instructions, natural user sequence toward a task, and use of function keys. Designers must make assumptions of user behavior and through prototype testing these assumptions can become refined and can be based on data. Significant savings

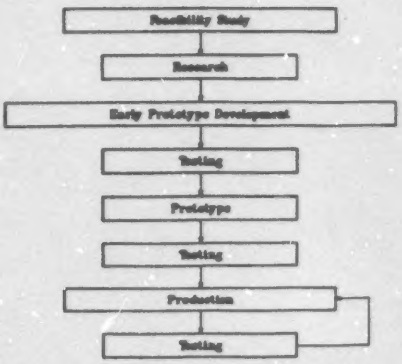


Figure 2. Human factors oriented system development.

can be realized at the debugging and maintenance stages if testing occurs during design as well as later stages (Gould and Lewis, 1983).

Iteration is essential. We often think of repeated testing and revision as merely fine tuning. Some consider that if a system was not perfect the first time it was tried then the designer was at fault. In fact, iteration is a necessary part of any development effort involving the creation of "friendly" user interfaces (Norman, 1983).

Users make mistakes. The word error in computing puts direct blame on the user, yet often user error is caused by an unforgiving user interface. Effective interfaces are designed under the assumption that what can go wrong will go wrong. System features such as spelling correctors and easy recovery from input not intended (e.g., in CBI, wrong multiple-choice option typed in by mistake) are no longer options for designers (Norman, 1983).

Users have limited memories. Often user error is caused by the user having forgotten instructions. People get interrupted, misplace printed documentation, use several systems and get them mixed up, etc. Users need to know where they are in a system from any screen, need to be provided with and know how to quickly access online help, and need to know how to get out of the system if they are interrupted. User interfaces can be designed with memory aids in mind. For example, consistent logical keys for obtaining online help, logical wording of commands (instead of numbers), and use of symbols (icons) instead of easily forgotten commands.

Users vary in their interface preferences. Experienced users prefer ease-of-use characteristics (e.g., command usage and capability to rapidly bypass certain screens), whereas

novice users prefer ease-of-learning characteristics (e.g., menus) (Yestingsmeier, 1984). There is no such thing as a "typical user," so in interface design one must examine the predominant preferences of a sample of the target population.

Users can be overwhelmed with information. In screen design, we often think of all that we need to cover and draw attention to. The user's perspective is different, however. Lists of more than seven, plus or minus two, can result in an information load. Continuous or frequent use of such features as blinking will eventually become self-defeating.

An expert opinion test is of limited use. Often in our design we go down the hall and consult a colleague to see if we are on the right track. Or we try to put on the hat of a novice ourselves. Experience has shown that our prior knowledge biases any testing or evaluation, and that, unless experts are the target population, obtaining expert opinions will not help us better match the user interface with the user's needs.

System messages need to be constructive. Often, when an error does occur, we are notified of this fact by the system but are not told how to fix it. Obviously, with complex functions it is difficult to provide all the necessary information on line. Many messages can, however, include brief, constructive suggestions. Messages such as, "error #17," "invalid input line," "NO," "SORRY," and "WHAT?" are discouraging to receive. Messages such as, "Only options listed above are acceptable to enter here" or "Command name must be followed by a space and a filename," quickly enable the user to input correctly without having to hunt through documentation or feel stupid (Dean, 1983; Shneiderman, 1982).

IV. WHAT CBI HAS TO OFFER USER INTERFACE DESIGN RESEARCH

As mentioned earlier, courseware design has for two decades focused on effective user interfaces. This has been necessary not only because the primary users of the courseware were not computer experts, but also because of the educational function of the systems. The following are specific areas in which CBI has obtained extensive experience, yet sometimes this experience has not been well documented in a form usable by the user interface design community.

Knowledge-based answer judging. Courseware developers have had to provide answer judging based upon actual or hypothesized knowledge of user behavior. Most often, the development team would hypothesize about student answers to short answer questions and use these in the coding. As students used the course and online data was analyzed, the answer judging was expanded to respond to a wider range of student input. Providing for constructive responses to variable user input is a new challenge to many computer systems people. They could benefit from learning more about flexible answer judging in CBI.

Online monitoring of user input. Historically, most courseware projects include the collection of user input on line. Many authoring systems have this feature by default, although collection is somewhat awkward with microcomputers. In non-educational software, the use of online monitors is new. Currently, an active focus of study in interactive software design involves how to analyze the mass of user input data collected so patterns of user behavior can quickly be identified. Because of the experience obtained in CBI we would make a contribution if successful implementation of courseware data analysis methods were published.

Friendly use of function keys. Because of the constant recognized need for such functions as back and help in CBI, users have, for many years, benefited from accessing these functions through easy-to-remember ways (e.g., PLATO's help and back keys).

Different pathways for variable user levels. The use of the pretest in much courseware allows experienced users to rapidly bypass information that is not needed. Some CBI courses use nonthreatening pretests (e.g., questions about experience) that conceptually may be transferable to other types of software.

Screen design tailored to user characteristics. The use of graphic representation, limitation of the amount of text on screens, and frequent interactivity are standard in CBI screen design. These characteristics are not easily nor commonly used in much non-educational software. Heines (1983) wrote a book on screen design for computer-assisted instruction containing examples that are useful elsewhere.

Formative evaluation methodologies. Consistent with the user interface design principle mentioned for early user testing, formative evaluation methods used in CBI testing are well developed. Such methodologies are emerging from the human factors field for non-educational software, yet documentation of the use of methods used in CBI (e.g., systematic observation of users) could contribute to this emerging effort.

V. WHAT USER INTERFACE DESIGN HAS TO OFFER CBI RESEARCH

Early courseware was designed around Skinnerian educational theory, with drill and practice being the predominant method of interaction. New teaching methods, such as guided-discovery and problem solving strategies, have emerged, based on the work of Piaget and others. Limitations of CBI authoring and resources have, however, prevented widespread development of courseware reflecting these newer theories. One example that does reflect the newer theories and has become extremely widespread is Seymor Papert's LOGO. There are, however, ways (in addition to obtaining such packages as LOGO) that courseware designers from small institutions with limited funding can build on research into cognitive modeling and friendly interface design. This would hopefully result in the creation of systems that are more precisely suited to their own students' needs.

Separate the user interface from the knowledge base. Both in development and in software coding, the separation of the user interface from the database of information provides several advantages. The database can be built by a separate team concurrently with the user interface, or the interface tool can be obtained from a vendor. The database can be updated without it affecting the interface. The user interface can be used with more than one database once it is built. Building such user interface management systems (UIMS), although labor intensive initially, is increasing (e.g., Buxton et al, 1983). This concept is behind the creation of some forthcoming authoring systems. But the idea does not yet have widespread use among the CBI community.

Integration of training to avoid mode change. Any change in mode required by a user solving a problem interferes with the thinking process. Users are more motivated to learn for a specific purpose for a capability not yet needed. Many friendly systems require almost no training (e.g., the Macwrite word processor on Macintosh), whereas others attempt to provide user-training through online help screens that can be browsed through. Online documentation can help to train users (Stoddard, 1984). Intelligent Tutoring Systems (Sleeman and Brown, 1983; Soloway, 1982) allow the user to be tutored as needed while solving a problem. Yet much CBI still occurs during conventional training; user motivation is sometimes low because the need for the information is not yet clear. Familiarity with the design of systems where training is not thought to be necessary and systems where training is embedded could aid us in our quest to maximize learning and minimize time invested in training.

Establishing a model of user behavior. Because the necessity for an in-depth, accurate model of user behavior is inherent in the creation of intelligent user interfaces, we can benefit from the study of the methodologies used to obtain detailed user behavior information (e.g., Kintsch et al., 1984; Card, Moran, and Newell, 1983).

Use of existing user interface guidelines. Although no standard for user interfaces has been established (Meads, 1984), several reports have been written containing specific guidelines for interface design (Smith and Aucella, 1983; Williges, 1981).

Learning styles and the stages of the computer user. The success of a system is dependent on how well the user interface matches the user's needs. As such, some user interface research has focused on the stages learners go through (e.g., Mozeico, 1982) and on characteristics of the learner (e.g., Lewis and Mack, 1981). Approaches used by researchers in these studies, which focused more on system implementation than on cognitive science research, have direct transferability to work in computer-based instruction.

REFERENCES

Buxton, W., et al., Towards a Comprehensive User Interface Management System, Computer Graphics 17 (3), SIGGRAPH '83 Conference Proceedings.

Card, S., Moran, T., and Newell, A., 1983, The Psychology of Human Computer Interactions, Erlbaum Publishing Company.

Dean, M., 1983, How a Computer Should Talk to People, IBM Systems Journal 21 (4), pp. 424 - 453.

Gould, J, and Lewis, C., 1983, Designing for Usability -- Key Principles and What Designers Think, ACM SIGCHI Proceedings, pp. 50 - 53.

Heines, J., 1983, Screen Design Strategies for Computer-Assisted Instruction, Digital Press.

Kintsch, W., Miller, J., and Polson, P., 1984, Method and Tactics in Cognitive Science, Lawrence Erlbaum Associates, Publishers.

Lewis, C., and Mach, R., 1981, Learning to Use a Text Processing System: Evidence from "Thinking Aloud" Protocols, ACM Proceedings, pp. 387 - 392.

Meads, J., 1983, SIGCHI/HFS Standards Meeting Minutes, ACM SIGCHI Bulletin 16 (1), p. 3.

Moran, T., 1981, An Applied Psychology of the User, Computing Surveys 13 (1), pp. 1 - 11.

Mozeico, H., 1982, A Human/Computer Interface to Accomodate User Learning Stages, Communications of the ACM 25 (2), pp. 100 - 104.

Norman, D., 1983, Design Principles for Human-Computer Interfaces, ACM SIGCHI '83 Proceedings, pp. 1 - 10.

Shneiderman, B., 1982, Designing Computer System Messages, Communications of the ACM 25 (9), pp. 610 - 611.

Sleeman, D., and Brown, J. S., 1983, Intelligent Tutoring Systems, Academic Press.

Smith, S., and Ancella, A., 1983, Design Guidelines for the User Interface to Computer-Based Information Systems, DTIC Technical Report ESD-TR-83-122.

Soloway, E., Rubin, E, Woolf, B., and Bonar, J., 1983, Memo II: An AI CAI Programming Tutor, ADCIS Conference Proceedings, pp. 185 - 192.

Stoddard, M., 1984, Online User Assistance Systems: Can They Help Ease the Demand on Trainers?, Data Training, pp. 34 - 36.

Weinberg, S., 1982, Human Factors Guidelines: Designing Interactive Computer Applications, Control Data Corporation Technical Report.

Williges, B., and Williges, R., 1981, User Considerations in Computer-Based Information Systems, ONR Technical Report.

Yestingsmeier, J., 1984, Human Factors Considerations in Development of Interactive Software, ACM SIGCHI Bulletin 16 (1), pp. 24 - 27.

Adaptive Testing by Computer: Benefits, Costs and Issues

J. Jon Veloski, Jefferson Medical College

Andrew R. Baggaley, University of Pennsylvania

ABSTRACT

This paper describes the technology of computerized adaptive testing and its history. In this form of administering tests, the items of a standardized examination are selected from a pool or bank of items according to an algorithm based on the psychometric properties of the items. Items are administered to each examinee according to the examinee's estimated level of ability related to what the test is designed to measure. When a predetermined level of accuracy or number of items have been reached the computer stops to test and provides a score. Thus it is likely that each examinee will respond to different test items, given a sufficiently large pool. Computerized adaptive testing promises many benefits, including improvements in the precision and validity of measurement. Certain costs, however, in addition to investment in computer hardware have restricted its use. New approaches to testing and measurement, such as latent trait theory, must be developed in order to make effective use of adaptive testing. Although the availability of inexpensive microcomputers will lead to increased interest in computerized adaptive testing, their use will not necessarily provide solutions to some of the problems of adaptive testing. Other issues related to the very nature of adaptive tests and the resulting scores may interfere with their implementation. Nevertheless, there are presently opportunities in some areas for the effective use of adaptive tests.

Introduction

Computer costs -- if they have not already done so -- will come down to the point where computer-based adaptive testing is economical. When this happens, adaptive testing will come into wide use.

> Frederick M. Lord at the First Conference on Computerized Adaptive Testing, June, 1975.

If the aircraft industry had evolved as spectacularly as the computer industry over the past 25 years, a Boeing 767 would cost $500 today, and it would circle the globe in 20 minutes on five gallons of fuel. .... The cost of computer logic devices is falling at a rate of 25 percent per year and the cost of computer memory at the rate of 40 percent per year.

> Hoo-min D. Toong and Amar Gupta in "Personal Computers", Scientific American, December, 1982.

Considering the predictions made during the 1970's by respected authorities in the field of educational testing and the radical changes that have been seen in the economics of computers over the last ten years, one might expect to find widespread uses of computerized adaptive testing today. In fact no large-scale operational uses of adaptive testing are visible in 1984, although several applications are being developed for selection of personnel in the military (Green, Bock, Humphreys, Linn & Reckase, 1982). There does continue to be an active body of research

literature on adaptive testing (Green, 1983; Vale, 1981; Volans, 1982). However, the gap between theory and implementation leads to questions about the relative advantages and disadvantages of computerized adaptive testing. Given the facts about the declining cost of computers, it seems possible that in addition to cost, other barriers might explain the apparent reluctance of professional testers to utilize computerized adaptive testing.

Computerized Adaptive Testing

In computerized adaptive testing, the items of a standardized examination are selected from a pool or bank of items and presented to each examinee by a computer. Testing often begins with the selection of an item of medium difficulty or an item matched to the ability of the examinee based on the examinee's previous performance in the battery of tests. The examinee's responses are recorded by the computer, which, after each response, calculates a score or revised estimate of the examinee's ability. This score is used by an algorithm, or set of rules stored in a computer's memory, to make decisions about the level of difficulty of subsequent items or tests that will be retrieved from the pool and administered to the examinee. Testing continues until a predefined criterion, such as a specific value for the standard error of measurement, has been reached. Throughout many years of research, other methods quite similar to adaptive testing have been referred to by the names of tailored testing (Urry, 1977), individualized testing (Lord, 1974), sequential testing (Linn, Rock & Cleary, 1972; Weitzman, 1982) flexilevel testing (Lord, 1971) and locator testing, most of which have been based on the same general approach. It is important to differentiate between adaptive testing and those forms of computerized testing concerned with the mere presentation and scoring of items by a computer (Johnson, 1982; Brzezinski, 1984). In the latter approach to testing by computer, one or more forms of conventional paper-and-pencil tests are stored in a computer so that they can be administered and scored interactively. The unique and essential characteristic distinguishing adaptive testing techniques from simple testing by computer is that, in adaptive testing, an algorithm guides the sequence of items to be presented to the examinee. In this way the examination is "adapted" or "tailored" to the unique characteristics of each examinee, with the expectation that measurement will be improved. Given a sufficiently large pool of questions that measure a single domain, each examinee will be presented with a different set of items tailored to his or her estimated level of ability. In this paper the term adaptive testing will be used to refer to the family of testing techniques that uses some sort of algorithm to select the sequence of items to be presented to each examinee.

The notion of an adaptive test can be traced originally to the work of Binet, when in 1909 he reasoned that if an examinee responded correctly to an item in an intelligence test, it would be most appropriate for the examiner to follow it by administering a more difficult question. Conversely, an incorrect response from an examinee would indicate a need for a less difficult question. This approach to measurement is analogous to accepted methods in psychophysics where the objective of measurement is to identify a threshold level of performance. Testing of response times, hearing ability, eyesight, and human memory has generally made use of this up-and-down branching method.

It has been unfortunate for the advancement of adaptive testing that the field of educational and psychological measurement since Binet has been dominated generally by requirements to test large numbers of examinees in scheduled group administrations. Bureaucratic organizations -- such as large universities, professional certifying groups, and governments -- began to use tests for decision-making in areas such as selection, counselling, and grading. In particular between World War II and the late 1970's, the capability to process large batches of examinations using mainframe computers and optical mark readers made the administration and scoring of standardized, group-administered paper-and-pencil examinations extremely economical thus limiting interest in adaptive testing. Nevertheless interest in adaptive testing was sustained during this period by Lord (1971) and others at the Educational Testing Service, as well as by testing experts in the government and military. A more complete list of the persistent investigators during this period is provided by Green (1983) who has also maintained activity in this field at the Johns Hopkins University. Advances in microprocessing technology in the 1980's are also inspiring renewed interest in adaptive testing (Green, 1983).

Interest in computerized adaptive testing has been increasing simultaneously with the expanding field of item response theory (Hambleton, 1983; Hambleton & DeGruijter, 1983). Referred to as latent trait theory and item characteristic curve theory, this alternative to classical true score theory is well-described in the

article by Hambleton, and is reviewed by Baker (1977) in the context of new developments in item analysis. Item response theory is directly applicable to computerized adaptive testing since it enables one to estimate the psychometric parameters of items without making assumptions about a particular population of examinees. The fundamental goal of adaptive testing, of course, is to make measurements of individuals without regard to the manner in which they are sampled from a population. A general familiarity with item response theory is needed to understand the operation of most computerized adaptive tests, and to evaluate alternative approaches (Bock, 1982; Cliff, Cudeck & McCormick, 1979; Cudeck, McCormick & Cliff, 1980; Nisbett, Adams & Arther, 1980).

The Benefits of Adaptive Testing

Although the highly developed paper-and-pencil tests in use today seem to do their job rather well, there are opportunities for improvement. A limitation of all tests of fixed length is that the scores are less precise at the extremes, since the majority of items in the test are designed to make discriminations among examinees in the middle of the distribution. In an ideal test, the error of measurement should be quite similar at all levels of examinee performance. In situations where scores on a test are used for decisions on the acceptance or rejection of examinees, it might be preferable to obtain a smaller error of measurement in the neighborhood of the cutting score. Computerized adaptive testing promises to alleviate these problems by selecting items that are most appropriate to the estimated level of ability of each examinee, thus deriving the maximum amount of information from each response. The precision of each examinee's test score can be monitored dynamically by a computer, and the desired level of precision can be attained by administering additional questions if necessary.

When computer technology of high quality is used for the administration of adaptive tests, it is likely that the clerical errors made by examinees will be minimized or even eliminated. Although many examinees have become accustomed to the use of optical scanning input sheets, the chance of error always exists. Advanced technology has only minimized the possibility of dirt, smudges and stray marks resulting in inaccurate scores, not eliminated them and transcribing errors are still possible. In adaptive testing, an examinee's response could be displayed by the computer for verification by the examinee before being incorporated into the score. In addition, small savings in costs associated with handling and scoring answer sheets can be realized, since the data are stored magnetically in the course of administering the examination.

It is possible that computerized adaptive testing will stimulate the development of new types of test items. For example, early evidence suggested that the number of attempts to reach a correct response might be a more reliable measurement than the commonly-used measure of whether or not a correct answer was given on the first attempt. This method of testing has not been widely used over the years because of the high cost of implementing it on paper-and-pencil examinations (Green, 1983). There might be an opportunity to use sentence completion or fill-in-the-blank items since a microcomputer could accept an examinee's free-response input from the keyboard, resulting in improved measurement in domains where distractors for multiple choice items are difficult to write. Other parameters which might be measured might include the amount of time required to read a paragraph or the elapsed time between the presentation of an item and the examinee's response. In some situations, there may be significant advantages in being able to administer examinations on a very flexible schedule in order to accommodate the availability of examinees. Adaptive testing will support the decentralization of testing sites.

Closely related to the advantage of flexibility in times of administration is the potential to provide immediate feedback to examinees and the decision-makers who use test scores. In the case of batteries of tests, it may be possible to save time for all concerned if scores on the initial tests are available immediately. For example, an applicant who scores poorly on the verbal scale of the Scholastic Aptitude Test (SAT) may be advised not to take certain advanced placement tests. Conversely, the student who scores high may choose to take additional tests. The examinee who performs poorly because of temporary physical illness or discomfort may be encouraged simply to return the next day and try for a higher score. The cost of repeated attempts as well as the time involved would probably limit any tendency for examinees to continue incessantly trying for a higher score, but the potential impact on some examinees of such a dilemma will be discussed later.

Many advocates of adaptive testing assert that an adaptive test of a given level of precision will, on the average, require less time to administer than an

equivalent paper-and-pencil test. By using an algorithm which can select only those items matched to an examinee's estimated level of ability and by eliminating the handling of answer sheets, time will be saved. Whether this benefit will be realized is debatable and will be discussed later, however, if a test does require less time to complete, it is possible that additional domains can be examined within a given testing time, resulting in greater validity in the entire battery of tests.

The Costs of Adaptive Testing

As suggested in the introduction, some advocates of computerized adaptive testing have frequently emphasized the cost of computer power as the most significant component in the cost of implementing this new method, and thus, the most important barrier to be overcome. A more careful inspection reveals that there are other costs, some of which are indirectly related to the computer, and others of which are much more difficult to identify and quantify.

At the present time the least costly component of a computerized adaptive testing system seems to be the digital computer or the cost of processing information during the administration of the test. Unfortunately, peripheral equipment for a computer -- such as visual display units of high quality, reliable printers, high-speed disc-storage units of large capacity and input devices such as carefully engineered keyboards -- are still relatively expensive, and they have not experienced the dramatic decreases that have been apparent in the prices of microprocessors. A thorough description of the specifications needing to be met in the design of equipment suitable for adaptive testing is given by Green et al (1982). High standards of reliability, human factors engineering to develop special input devices, and clarity of visual displays will demand prices for equipment far in excess of a few thousand dollars for personal computers advertised in newspapers and magazines. Naive observers claim that computers are now inexpensive and that the time has arrived for widespread use of adaptive testing, however machinery required to meet the high standards of large professional testing organizations still represents a huge capital investment.

Nevertheless, a more substantial investment may be required to develop an adequate item pool from which item selections can be made. To some extent the item pools already available in large testing groups may be of value; but certain limitations need to be recognized. Some types of items may need to be rewritten, redesigned, or even discarded. For example, paragraph comprehension items are not well suited to present approaches to adaptive testing. Typical computer screens can display between 20 and 66 lines. Even the largest available display would be inadequate to handle many of the paragraphs used in examinations such as the SAT. Existing paper-and-pencil examinations make it possible for the examinee to use various strategies such as reading the items before reading the paragraph, underlining key phrases in the text, or rereading a paragraph while responding to items. These techniques may not be possible when using a computer and complex technologies may be required to permit the examinee to page back and forth easily. It would seem that such changes would alter the psychometric characteristics of the items. More importantly, the item response models being used require a strict assumption of independence among items, which could not be satisfied when multiple items refer to the same reading sample. Of course each reading passage could be treated as an item unto itself. It could be argued that new types of reading passages need to be developed, but it would be essential to consider the effect of their substitution on the content validity of existing examinations. Closely related to the dilemma of paragraph comprehension items are those types of items dealing with interpretations of figures, maps, and other pictorials. Here, the visual quality of widely available computer display devices will be one limiting factor, but issues of organization and timing similar to the paragraph items are apparent. Green et al (1982) point out that in order for adaptive tests to operate efficiently, items with superlative discriminating power are required. For example, point-biserial correlations in excess of .80. The use of items of lower discriminability will cause instability in the successive estimates of an examinee's ability and will result in errors in the selection of items. If only poor items were available a computerized adaptive testing algorithm might fail to converge on an accurate estimate of the examinee's ability. Whether existing items are redesigned or replaced with new and equally valid types, it seems that the costs will be significant to develop and calibrate items of high quality.

Many people will have difficulty in grasping the underlying psychometric theory of adaptive testing. Questions will be asked, and undoubtedly complaints will be voiced by the public, as well as by those who make use of test scores. How

will it be possible to explain succintly why two people could answer correctly about one-half of the questions in a test, yet one would receive a passing score while the other would fail? Those who led the truth-in-testing movement of the early 1980's would find new opportunities to assail the credibility of standardized tests. Test disclosure would be a complex issue as each examinee would see a different set of items. Furthermore, such inquiries could even lead to questions about the equating of parallel forms of tests which has been faithfully accepted by the public. The cost of educating the users of information from adaptive tests may be significant. Baker calls attention to the fact that, in 1977, discussions of item analysis in textbooks on measurement were still based on the classical psychometric model rather than the use of item characteristic curves on item response theory. He states that a shift in emphasis would be needed to produce a new group of consumers of test information who would understand the operation of new psychometric approaches such as adaptive tests.

Issues

A set of concerns which cannot be identified as direct and tangible costs of implementing adaptive testing recurs in the literature. Nevertheless the potential effects must be considered. At the present time standardized tests such as the SAT seem to work fairly well in that they have been shown to be highly reliable and somewhat valid. There may be significant resistance by users of test information to change methods unless the benefits are perceived to exceed the costs. It should be remembered that the development of adaptive testing has been highly theoretical and that a large portion of the published research has been based on high level mathematics and complex computerized simulations rather than empirical studies with real examinees (Bock, 1982; Cliff et al, 1979; Ree, 1981). The users of test information may be skeptical about the utility of the scores and not readily accept them.

In some areas such as colleges and universities, current forecasts indicate a reduction in competition for admission which would seem to imply a reduced demand for improvements in testing technology. Predictions of reduced college enrollment may imply less need for elegant standardized tests such as the SAT as many colleges try to recruit applicants and in effect operate with open admission programs. Similar trends are emerging in law schools, dental schools and medical schools. The largest consumers of test information may be unmotivated to pay for the cost of developing adaptive tests.

The accuracy of adaptive testing is dependent on some rather crucial assumptions. First is the accuracy of the estimates of item difficulty derived from calibration procedures. If these estimates of difficulty are not accurate, items selected for an examinee may be inappropriate, and errors in subsequent selections will be compounded. It is recognized that some calibration procedures are not robust with regard to the violation of statistical and psychometric assumptions, and research continues on the actual consequences of these violations. The unidimensionality of items in relation to the ability being measured is critical. If the domains measured by items vary at different levels of difficulty, then examinees at these different levels will be tested on different dimensions. For example, the most difficult items in an achievement test may also measure reading ability.

New issues of differential validity may arise with regard to the scores derived from adaptive testing. Adaptive testing promises to increase the precision of measurement at the top and bottom levels of performance, but it is not clear how these changes might affect the predictive validity of test scores. When subgroups within the population differ in ability, changes in differential prediction may be observed. Traditional methods of identifying ethnic, cultural, and sexual bias in items will not be applicable since these have been based on classical test theory. New procedures will be needed to handle the case where all examinees do not see all items.

The concerns of the public may stimulate legal controversy if tests tailored to the abilities of examinees are misinterpreted as being discriminatory. Green (1983) dismisses this complaint with an analogy to athletic events where a jumper who cannot clear a three-foot hurdle does not have a legal right to attempt a four-foot hurdle. However, in our society testing plays a central role in many decisions involving education and careers, and any suggestion of arbitrariness will be attacked.

It has been suggested that adaptive techniques will produce ideal power tests since each examinee will be able to work at his or her own pace. However, a computer has the potential to create an illusion of speediness. Examinees who are accustomed to the pace of commercial television and video games may behave differently than they would if they were working with a traditional test booklet. A properly

functioning adaptive test, which selects items of appropriate difficulty and high discrimination, may actually be more stressful for examinees than conventional tests, which, to the average examinee, include items that might range from being quite simple to virtually impossible. There is controversy among experts as to whether examinees should be allowed to omit a response in an adaptive test. Some argue that once an item has been selected by the computerized algorithm as the best discriminator at the current level of estimated ability, a response from the examinee is needed in order that the system work efficiently. From the examinee's point of view it may be discomforting to be forced to take a wild guess if the answer is not readily apparent. First experiences with adaptive tests may provide unpleasant experiences for all but the most motivated examinees. Policies would need to be established as to how many times an examinee would be allowed to repeat an adaptive test.

Finally, if item banks are decentralized in many small computers new issues of test security may arise. Many small computers throughout the country containing the item banks for competitive examinations such as the SAT might be vulnerable to theft. An unscrupulous test preparation firm might be able to strengthen their examination review course by having access to the entire item bank. If one were stolen the credibility of a major examination might be weakened.

Conclusions and Implications

The allure of computerized adaptive testing is no different from that of other new technologies. There are appealing and somewhat seductive promises of increased effectiveness, apparent opportunities for long-term cost reductions based on a modest capital investment and, initially, only a few conspicuous disadvantages or adverse side effects. Careful inspection suggests that there are several unresolved psychometric questions which presently impede the implementation of adaptive testing of a quality to meet the fairly high standards of existing paper-and-pencil tests. It is surprising to find that there are so many unresolved questions about the methods to be used, and of course there is little evidence of systems being implemented. The cost of computer technology appears to be a minor constraint when one considers the psychometric difficulties. Nevertheless, present trends suggest that the military will introduce adaptive testing before the end of this decade. The advantages of security and speed of turnaround are perceived to outweigh both the costs of development and any issues related to acceptance by examinees. Historically, the testing methods of the military have ultimately influenced public and private testing groups. Therefore, if their implementation is recognized as being successful it seems likely that others will follow.

Practitioners such as educators, psychologists, and administrators who must make decisions based on information provided by adaptive tests must learn to become intelligent consumers. This will require an understanding of the fundamentals of item response theory and the methods used to calibrate items. Initially, the economy and allure of computers may produce inferior adaptive tests. Practioners must be able to scrutinize these products, identify weaknesses, and choose or reject them accordingly.

For research, the introduction of computerized adaptive testing will present much virgin territory. As has been suggested, there is continuing controversy about the effects of specific methods of testing. What are the real effects of the use of various methods of item calibration? Should the stopping point be determined by the standard error of measurement alone, or should it also take into account the number of items administered, or perhaps be constrained by duration of time? What is the effect of allowing examinees to omit items, or the consequences of prohibiting omitted responses? But more importantly, the scores produced by adaptive tests are so fundamentally different that their validity will need to be examined. If a test such as the SAT were to be replaced with an adaptive test, then all earlier validity studies of the SAT would need to be scrutinzed. A score derived from a paper-and-pencil version of the SAT would be quite different from a score based on a computerized adaptive test. On the one hand, this sounds like a windfall for statisticians, psychometricians, and others interested in educational research, but, of course, one must ask who will be willing to pay for all of this research?

References

Baker, F. B. Advances in item analysis. Review of Educational Research, 1977, 47, 158-178.

Binet, A. Les idees modernes sur les enfants. Paris: Ernest Flammarion, 1909.

Bock, R. D. "Adaptive EAP estimation of ability in a microcomputer environment". Applied Psychological Measurement, 1982.

Brzezinski, E.J. Microcomputers and testing: where are we and how did we get there? Educational Measurement: Issues and Practice, 1984, 3(2), 7-10.

Cliff, N., Cudeck, R., & McCormick D. J. Evaluation of implied orders as a basis for tailored testing with simulation data. Applied Psychological Measurement, 1979, 3, 495-514.

Cudeck, R., McCormick, D.J., & Cliff, N. Implied orders in tailored testing: simulation with the Stanford-Binet. Applied Psychological Measurement, 1980, 4, 157-163.

Green, B. F., Bock R. D., Humphreys, L. G., Linn, R. L., & Reckase, M. D. Evaluation plan for the computerized adaptive vocational aptitude battery. Baltimore: The Johns Hopkins University, 1982.

Green, B. F. Adaptive testing by computer. In R. B. Eckstrom (Ed.), Measurement, technology and individuality in education. new directions for testing and measurement. San Francisco: Jossey-Bass, 1983.

Hambleton, R. K. Developments in latent trait theory: models, technical issues and applications. Review of Educational Research, 1978, 48, 467-510.

Hambleton, R. K. & De Gruijter, D. N. M. Application of item response models to criterion-referenced test item selection. Journal of Educational Measurement, 1983, 20, 355-367.

Hambleton, R. D. (Ed.). Applications of Item Response Theory. Vancouver, British Columbia: Educational Research Institute of British Columbia, 1983.

Holmes, S. E. Unidimensionality and vertical equating with the Rasch model. Journal of Educational Measurement, 1982, 19, 139-147.

Johnson, C. W. Microcomputer-administered research: what it means for educational researchers. Educational Researcher, 1982, 11(10), 12-16.

Linn, R. L., Rock, D.A., & Cleary, T.A. Sequential testing for dichotomous decisions. Educational and Psychological Measurement, 1972, 32, 85-95.

Lord, F. M. Robbins-monro procedures for tailored testing. Educational and Psychological Measurement, 1971, 31, 3-31.

Lord, F. M. Individualized testing and item characteristic curve theory. In Kratz, D. H. et al (Eds.), Contemporary developments in mathematical psychology. San Francisco: W. H. Freeman, 1974.

Nisbet, J., Adams, M., & Arther, J. A simple form of tailored testing. British Journal of Educational Psychology, 1980, 50, 301-303.

Proceedings of the first conference on computerized adaptive testing. Washington: United States Civil Service Commission, 1976.

Rasch, G. An item analysis which takes individual differences into account. The British Journal of Mathematical and Statistical Psychology, 1966, 19, 47-49.

Ree, M. J. The effects of item calibration sample size on item pool size and adaptive testing. Applied Psychological Measurement, 1981, 5, 11-19.

Toong, H. D. & Gupta, A. Personal computers. Scientific American, 1982, 247(6), 87-107.

Urry, V.W. Tailored testing: A successful application of latent trait theory. Journal of Educational Measurement, 1977, 14, 181-196.

Vale, D. C. Design and implementation of a microcomputer-based adaptive testing system. Behavior Research Methods and Instrumentation, 1981, 13, 399-406.

Volans, P. J. Pros and cons of tailored testing: an examination of issues highlighted by experience with an automated testing system. International Journal of Man-Machine Studies, 1982, 17, 301-304.

Weitzman, R. A. Sequential testing for selection. Applied Psychological Measurement. 1982, 6, 337-351.

Wright, B. D. & Stone, M. H. Best Test Design. Chicago: MESA Press, 1979.

Measuring the Effectivenss of a Computer-Based Self-Assessment Program

J. Jon Veloski, James Robinson, Mohammadreza Hojat

Jefferson Medical College

ABSTRACT

Recognizing a continuing need to measure the effectiveness of computer-based learning programs a study of the results of one such program was conducted over an eight-year period. Students expected to have difficulty in passing a comprehensive examination in the basic medical sciences were invited to use a computerized self-assessment program during a six month period prior to the examination. The system, written in VSAPL on a mainframe computer, included over 1000 multiple-choice questions with explanations of correct and incorrect answers. A comparison of the performance of 151 students who completed the program with that of 60 students who declined the use of the program showed a 30 point difference favoring those who completed the program. A 29 point difference was found between the performance of the group who finished and a group of 70 students who began the program but did not finish. These results certainly have implications for educators and designers of computer-based instruction.

Introduction

Many applications of computer-assisted instruction, computer-managed instruction and computerized testing have been developed over the last two decades. Their success has often been judged solely on reports of student and faculty satisfaction. It appears that research on the effectiveness of many of these programs has been limited. Are these programs actually more effective than other approaches to learning? A recent article appearing in Science concerning the use of computers in education, reported that "80 percent of educational programs have gone on the market with little or no research or testing" (Holden, 1984). This paper describes an automated self-assessment program in operation since 1976 and designed to help second-year medical students prepare for a comprehensive professional certifying examination, Part I of the National Board Examinations. Data are presented based on eight years of research conducted to address the following question: Can a computerized self-assessment program improve selected students' performance on a comprehensive examination?

The Comprehensive Examination

Comprehensive examinations, such as professional certifying and licensing examinations, represent major hurdles to some students. In medicine, law or dentistry, whether or not a student is able to pass the examination often determines the future of his or her career. The predicament of academically marginal students provides an important opportunity for educators to design computer programs to assist these otherwise qualified students in preparing for examinations. Students in need of remedial assistance are often highly motivated, but busy faculty members may be reluctant to devote extra time to help them.

The development of the self-assessment program was supported in part by a grant from the William Penn Foundation.

Part I of the National Board Examination is a two-day comprehensive test of students' knowledge of the basic science areas (anatomy, behavioral sciences, biochemistry, microbiology, pathology, pharmacology and physiology). The examination consists of about 1000 multiple-choice questions. The majority of medical students take Part I at the end of their second year of medical school after having completed the relevant courses in the seven areas examined. Examination scores are standardized with reference to a number of years of examinees, and scores are reported on a scale having a mean of 500 and a standard deviation of 100 (Hubbard, 1978). The passing score is 380, so that about 11 percent of all examinees will fail over a number of years. Whether or not an examinee who has failed can repeat the examination is decided by the faculty of his or her medical college. At a number of medical colleges, such as the Jefferson Medical College, students must pass Part I in order to continue their medical education. At the same time, satisfactory completion of Part I and the companion examinations, Parts II and III, administered later in the career of the medical student qualifys him or her for licensure to practice medicine in nearly all states.

Although the failure rate on Part I at Jefferson has generally been below five percent, many members of the faculty have always been concerned about those few students who encounter difficulty. In 1975 a project was undertaken to develop a self-assessment program that would help academically marginal students prepare for the examination. The program was designed to avoid some of the pitfalls that continue to impede the effective uses of computers in education (Atkinson, 1972; Arons, 1984). First, it was decided that the program would help students identify their strengths as well as their weaknesses before trying to help them correct the weaknesses. The system would aid them in setting priorities for study. Second, the program was intended to be sufficiently flexible so that students could review material at their own pace without a rigid schedule. In general, it was determined that the program should be able to operate without impinging upon the crowded schedules of either the students or faculty (Kulik, Kulik and Carmichael, 1974).

It was decided that, to use students' time and computer resources most efficiently, the Self-Assessment Program would be offered only to those students predicted to encounter difficulty with Part I. To offer the remedial program to all students, including those with little chance of failure, might overemphasize the comprehensive examination and cause unnecessary anxiety among students who would probably do well. The goal of the program was established to improve performance at the lower extreme of the class, not to raise the scores of all students. Actuarial predictions of students' performance on Part I were prepared each year using multiple linear regression of previous years' data. Students whose predicted scores were within approximately two standard errors of estimate of the failing score of 380 were selected to participate in the remedial program. The variables used in these predictions and the cross validations during a ten year period are shown in Table 1. A scatter plot of the relationship between predicted and actual scores is shown in Figure 1. In general, it can be seen that the predictions have been very accurate and there has been little shrinkage in the validity coefficients from the regression equation to the cross-validations.

The Self-Assessment Program

The Self-Assessment Program was implemented in 1976 on a mainframe computer using five workspaces written in VSAPL (Gillman & Rose, 1979). One is a utility workspace that allows the clerical staff to create and maintain data sets for the storage of examination questions, without knowledge of the computer's operating system. A generalized text editor permits the clerical staff to add, modify and delete examination questions from the files for each discipline. A third "master" workspace is used by supervisory staff to add and delete student accounts from the system, and to monitor student utilization of the system in the fashion of computer managed instruction. Another workspace with locked functions is automatically loaded as each student uses the computer, to administer the self-assessment examinations and to maintain a record of each student's performance in the master file. The fifth workspace performs item analysis procedures and generates summary reports for the students and faculty.

Up to 75 students may use the program during any given time period. After being assigned an account on the mainframe timesharing system, a student may use the computer at his or her own convenience, subject to scheduling time on the computer terminals located in the university's library. Last year three students did use the computer program from their own homes using their own computers, terminal emulation programs and modems. Although a sequence of review of the seven areas is

recommended by the faculty, each student may progress through the 160 questions in each of the 7 areas at his own pace. Most require between 20 and 30 hours over a four month period to complete the entire program.

The system incorporates a set of unique features that distinguish it from many other approaches to computer-based self-assessment now in operation (Tait & Hughes). One is that a student may enter questions or comments about any item in the examinations using free text. These questions and comments are stored and reported to faculty members. The students' comments can be used by the faculty to improve the quality of the items in the self-assessment program and to understand better the weaknesses of students. The system also monitors the number of attempts that each student requires to answer a questions correctly as well as the amount of time needed to answer questions. The measurement of these parameters may be used for future psychometric research. The Self-Assessment Program contains other features that are not unique in 1984, but were considered innovative in 1976. All questions are stored and presented in upper/lower case which has led to greater acceptance by students than systems which use only upper case displays. The computer program used by the students is extremely resistant to tampering and there have been no reports of failure or breech of security in the eight year period.

If the system were being designed in 1984 rather than 1974 it is unlikely that a mainframe computer would be used for development. Today's microcomputers offer speed, high quality graphics and satisfactory storage capacity at a reasonable price. Nevertheless, a mainframe timesharing system has offered advantages. One is the convenience of a centralized data base. Corrections of typographical errors in items could be effected instantly so that students were administered the most complete and up-to-date questions. Such capabilities will be available in networks of microcomputers, but these are not yet widely available. Although security was not a prime concern in the project it was recognized that widespread dissemination of the items used in the self-assessment program would reduce their usefulness as tools for review. Mainframe computers with video display terminals provided security against casual printing of all of the questions by the students. Another advantage of the mainframe computer was that data on all students progress and performance were stored centrally and could be monitored efficiently. Otherwise, it would have been necessary to compile data from many floppy discs. Despite these advantages the program is being converted for operation on microcomputers in order to exploit their speed, availability and low cost.

Results of the Self-Assessment Program

Between 1976 and 1984 the Self-Assessment Program was offered to 303 of 1766 sophomore medical students at Jefferson. Table 2 shows the mean predicted scores on Part I for these students as well as their actual scores on Part I. Those students who accepted the offer to participate and completed all or nearly all of the items in the Self-Assessment Program realized an average gain of 30 points from their predicted scores to actual scores. These differences are statistically significant ($p<.01$) when subjected to analysis of variance. The few out-of-phase students who did participate in the program scored higher on Part I than those who did not finish, as well as those who declined.

Discussion and Conclusions

There is a need for objective evaluation of the effectiveness of all computer-based educational programs. In the present study a program was developed to raise the level of performance of selected students on a comprehensive examination, and eight years of data suggest that the goal was reached. A computer-based self-assessment program can raise the mean level of students' performance on the comprehensive examination. However, there are several limitations in the design of the study which must be considered. The ideal design would have been experimental, with students being randomly assigned either to the Self-Assessment Program or to a control group. It might be argued that those students who volunteered for the program were simply more highly motivated than those who did not, and their increased achievement on the test was a function of their motivation rather than the Program (Rosenthal & Rosnow, 1975). Although this may account for some of the difference, questionnaires obtained from students and informal feedback indicated that some of the students, who did not participate in our Program, chose other, equally demanding programs that are available, such as commercial review programs. It could be argued that the students who chose to use the Self-Assessment Program were <u>less</u> motivated that other students since they simply used a program that was provided by the faculty. A second limitation of the study is that the effects of the self-

assessment strategy and the use of a computer are confounded. The ideal study might have included a control group that was administered self-assessment questions by the paper and pencil method rather than by computer. This was not done in the present study because the costs to the faculty and the students of conducting two programs in parallel would have been impractical.

A finding of positive effects for a remedial program must be carefully examined due to the threat of regression effects. If individuals with scores below the mean are selected, one would expect their performance to increase whether or not the treatment is effective because of the statistical phenomenon of regression toward the mean. In the present study the use of two control groups with similar levels of predicted scores helps to dismiss this potential threat to validity. The students who began the program but did no finish had the lowest mean predicted scores and their actual mean was only one point higher.

It can be concluded that an automated self-assessment program can raise the scores of selected students on a comprehensive examination. These results have implications for educators who wish to use this approach. Since it is important to measure the effectiveness of all computer-based educational programs, the quasi-experimental method used in this study may be useful to developers of such programs.

References

Arons, A.B. Computer-based instructional dialogs in science courses. Science, 1984, 224, 1051-1056.

Atkinson, R.C. Ingredients for a theory of instruction. American Psychologist, 1972, 921-931.

Gillman, L. & Rose, A.J. APL: An Interactive Approach. 2nd ed. New York: John Wiley & Sons, 1979.

Holden, C. Can home computers transform schools? Science, 1984, 225, 296.

Hubbard, J.P. Measuring medical education: the tests and experience of the National Board of Medical Examiners. 2nd ed. Philadelphia: Lea & Febinger, 1978.

Kulik, J.A., Kulik, C.L. & Carmichael, K. The Keller plan in science teaching. Science, 1974, 183, 379-383.

Rosenthal, R. & Rosnow, R.L. The Volunteer Subject. New York: John Wiley & Sons, 1975.

Tait, K. & Hughes, I.E. Some experiences in using a computer-based learning system as an aid to self-teaching and self-assessment. Computer Education, 1984, 8, 271-278.

FIGURE 1

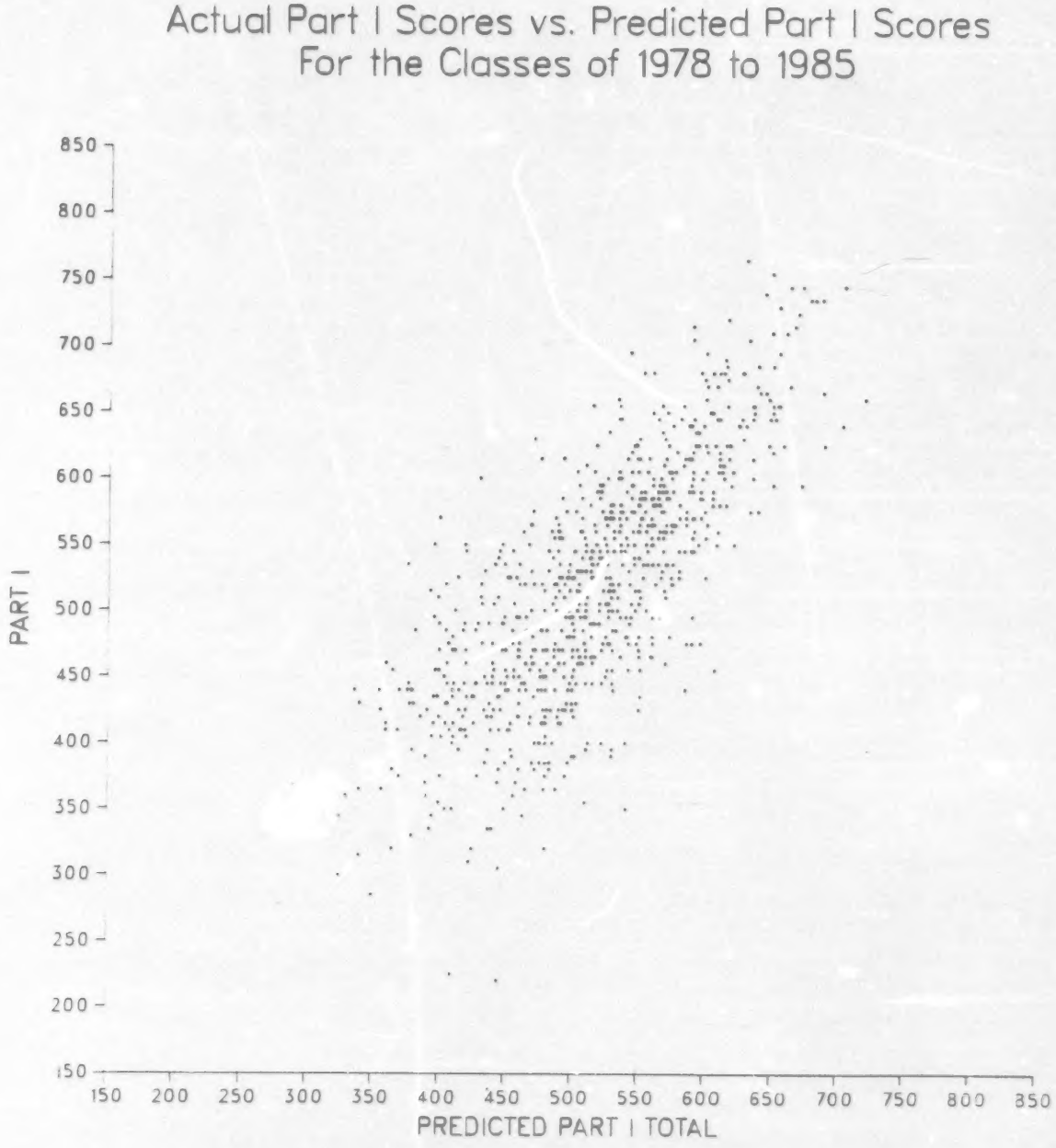

Actual Part I Scores vs. Predicted Part I Scores
For the Classes of 1978 to 1985
PART I
850
800
750
700
650
600
550
500
450
400
350
300
250
200
150
150 200 250 300 350 400 450 500 550 600 650 700 750 800 850
PREDICTED PART I TOTAL

Table 1 Results of Predicting Students' Scores on Part I of the National Board Examination

Academic Year	Actual Validity (R^2)	Expected Validity (R^2)	Variables Used in Predictive Equation
1983-84	.67 (.44)	.83 (.69)	Science Problems Reading Physiology Mechanisms of Disease
1982-83	.79 (.62)	.82 (.67)	Mechanisms of Disease Science Problems Physiology
1980-81	.70 (.49)	.74 (.55)	Mechanisms of Disease Physiology Science Problems
1979-80	.73 (.53)	.82 (.68)	Cell and Tissue Biology Function and Structure Mechanisms of Disease
1978-79	.73 (.53)	.85 (.72)	Cell and Tissue Biology Function and Structure Pathology Examination
1977-78	.84 (.71)	.82 (.67)	Cell and Tissue Biology Function and Structure Mechanisms of Disease
1976-77	.81 (.66)	.84 (.70)	Pathology Function and Structure Cell and Tissue Biology MCAT Science Sex
1975-76	.83 (.69)	.85 (.73)	Pathology MCAT Science Cell and Tissue Biology Sex
1974-75	.82 (.67)	.76 (.58)	Pathology MCAT Science Cell and Tissue Biology Sex

Table 2 Eight Year Evaluation of the Self-Assessment Program 1977-1984

	Mean on Part I (Total)					
	Predictions Available				Predictions Not Available	
	n	Predicted	Actual	Change	n	Actual
Students who completed all questions in the Program	151	421	451	+30	11	480
Students who began but did not finish	70	410	411	+ 1	6	428
Students who were offered the Program but declined	60	420	420	0	5	411
Other students at Jefferson	1425	535	537	+ 2	32	513

NOTE: Predictions were not available for those few students who wer out-of-phase because they had either failed a course or taken a leave of absence during their freshman year.

IBM PILOT - SOPHISTICATION AND SUPPORT

George Gerhold and Larry Kheriaty
Western Washington University
Bellingham, Washington 98225

Background

In the mid-1960's, John Starkweather, Marty Kamp, and friends designed an author language for use on small computers. By 1973, they had attracted enough users and implementations to justify a standardization process; the result was named PILOT 73. Although limited in power, PILOT 73 established a clean and simple core syntax and a suitable set of conventions for language extension. During the 1970's several extended versions of PILOT appeared, most of which made use of the higher power of the then current minicomputers and some of which preserved the clean, simple syntax. By the late 1970's it was possible to implement extended versions of PILOT on microcomputers. We established a defacto standard version, called COMMON PILOT, by writing interpreters for many of the most popular microcomputers. COMMON PILOT served as the basis for Apple, Texas Instrument, Radio Shack, Commodore, Monroe, Acorn, and the IBM versions, as well as for generic CP/M, PASCAL, C, and MS DOS versions. However, the graphic and color capabilities of microcomputers and the obvious utility of these features in instructional applications demanded further extension of PILOT if obsolescence was to be avoided. The versions listed above include many extensions, notably extensions for screen display. Unfortunately, in some implementations execution speed suffers, and many authors have produced courses that demonstrate possibilities but which are not appealing to impatient students. IBM PILOT represents a large step forward. The syntax has been refined and simplified; useful extensions to the language have been included; a mechanism for extending the language has been devised, and--for the first time--a development-support group for the language has been funded. In this paper we describe these changes: syntax refinement, additions to the language, mechanisms for extensions, and the development-support group.

Syntax Refinement

An author language should be as simple as possible; once authors have a minimum experience with the language their guesses as to how to code actions usually should be correct. Guesses as to syntax are more likely to be correct when there are no special cases. For instance, if an expression can be used in place of a number in any one place, then the syntax will be simpler if an expression can be used in place of a number anywhere a number makes sense. If string variables can be subscripted in any one situation, then they should be subscriptable anywhere string variables might appear. Those of you familiar with PILOT know that these two examples were chosen intentionally; in many versions it is not possible to use expressions in subscripts, nor is it possible to imbed subscripted string variables in Type commands. The ideal, which IBM PILOT approaches, is that expressions, variables, and numbers can be used anywhere that any one of them can be used, and a similar flexibility for use of string expressions, string variables, and literals is desirable. IBM PILOT does have some limitations on use of expressions and string variables within one command type (the TS: command), otherwise the ideal interchangeability is achieved. Use of expressions as subscripts is

especially powerful--saving commands. variables, and time. Use of expressions, including subscripted variables, within the text of Type commands also simplifies many programs. Another class of special cases has been eliminated; system variables can be used exactly like other variables, one can include a subscripted portion of the answer buffer within the text of a Type command.

CAI authors often have difficulty coping with the distinctions between string values and numeric values (e.g., between the character 1 followed by the character 0, as distinguished from the number ten). In fact, these distinctions are almost never of value in CAI. Therefore, IBM PILOT (and some of the other versions listed above) does not require the author to handle such distinctions. The computer automatically makes the appropriate conversions between strings and numbers from context. Many authors now use this feature to store numbers as a list in a string instead of using numeric arrays. For example, the command

```
C: A$(3,4) = B$(1,4) + 2*B$(8,4)
```

is legal.

Microcomputer versions of COMMON PILOT include several special string editing operations: capitalization, removal of particular characters, switching of pairs of characters. In early versions these operations were made part of an alternate form of the Compute command, thus creating another special case in the syntax and restricting them to operations on a complete string. In IBM PILOT these operations are string functions. This means that they can be made part of complex string expressions, that they can be used in combinations and on substrings, and that the Compute command is always an assignment command.

Additions to the Language

There are too many additions to the language for us to give a complete list here. Obviously as the graphic capabilities of the machine increase, additional syntax must be added to control the capabilities. Most notable among these features are multiple windows and multiple modes. The latter is an area of continuing concern, as authors want to write programs which will make use of the capabilities of the professional display but still function in monochrome and every level of graphic display between these two extremes.

User-defined characters are an essential feature in any author language. In IBM PILOT the characters are defined via PILOT commands as part of the program, not via a separate character editor. There are 128 characters which can be added to the standard ASCII set (other versions of PILOT can only replace standard characters). While it is a great advantage to have these extra characters, there must be a way for the student to access them easily from the keyboard. Fortunately, IBM allows redefinition of the ASCII code generated by any keystroke, so IBM PILOT has a command for this redefinition. Any particular Accept command can be modified either to use or to ignore the redefinition. Thus, successive student responses can alternate between, say, Russian and English without repeated reprogramming of the keyboard.

Most versions of PILOT access files via records of fixed length. String operations are then used to construct or decompose the fixed length records. IBM PILOT allows the author to begin reading files at any byte position. The length read is set by the dimension of the string variable the information is read into. Files are written in corresponding fashion. If, as is usually the case, files are used mainly for storing student records, then this change is not important (there is a separate student record command). If files are used to store data which is to be used in the program, then this change makes access much easier.

As CAI coursework becomes more sophisticated, the need to access external devices increases. Examples include video tape, video disks and speech generation boards. Unfortunately, there is a lack of standardization in the interfaces, so the best we can do is to provide a variety of hooks which can be customized. IBM PILOT includes mechanisms for dropping into assembler, for reading and writing bytes to specific memory positions, and for communicating with the RS232 port.

PILOT programs are written and edited with a text editor. Although many text editors will work, most text editors do not handle the user defined characters that many authors will want to use. Therefore, a simple editor which will handle ASCII characters numbers 128-255 is included. Complex graphic screens are best created with one of a variety of graphic editors. IBM PILOT can display any screen which has been stored as a bit-map image, so most graphic editors can be used. In many cases a series of editors has been used to create a combination of effects. The graphic command in PILOT is used to create simple drawings or additions to graphic screens.

Mechanism for Extensions

CAI is in a state of flux; new ideas for applications demand new capabilities in author languages, and new peripherals demand new interfaces. IBM PILOT provides a mechanism for adding new, customized features to the language without requiring a new version or an update of the language. The V command (V for video, the most obvious use) takes the contents of any designated string variable and executes these contents as machine code. Therefore, to add a new feature to IBM PILOT one needs to first write and assemble the code and store it as a binary file. Within PILOT one loads the code into a string variable via a File In command and executes it when needed via a Video command.

While the process is straightforward, it requires programming skills beyond those of the average CAI author. Therefore, we have begun producing and distributing a set of useful additions as as Advanced Feature Library. The first item in this library is a fast graphics routine, SQZ. SQZ compresses a screen image and stores it together with the code to redisplay it in a binary file. These compressed images occupy significantly less disk space than screen images, somewhere between 1/10 and 1/2 depending on complexity. Each compressed image is read into a separate string variable and displayed, when needed, with a V command. The actual display process requires about 1/10 of a second, so it is much faster than displaying a standard screen image.

A binary program which allows execution of DOS commands is included in the AFL. The actual DOS command is imbedded in the contents of a string variable via a PILOT Compute command and then executed via a Video command. While standard DOS commands like FORMAT and DIR can be executed, the most useful applications are calls to programs in other languages like PASCAL, BASIC, or C. When the program called returns control to DOS, control is passed immediately to the PILOT command which follows the call, so in effect one has a subroutine call to any programs available from DOS. There are some limitations; both PILOT and the called application must fit in memory, so you must have at least 256K available, and parameters can be passed only via files. This utility opens a whole spectrum of new applications for PILOT. Instructional programs which require extensive number crunching can be written in PILOT with another compiled language used for the heavy computations; PILOT can be used to write front ends and tutorials for other software packages.

In some applications the author would like the student to edit a response (or other text) instead of retyping a new one. A routine which provides a simple student editor for use within a PILOT program is included. This has been used in tutorials on programming (the student edits a multiline program) and in tutorials on English composition (the student edits paragraphs).

One of the useful features in PILOT is the escape feature. If the escape feature is enabled, then every student response is automatically checked for the presence of the @ character in the first position. If present, an automatic subroutine call is executed. The AFL provides a routine so that the author can provide as many keys as desired for various special actions. The obvious choice is to use the function keys for this.

The above examples give an indication of what is possible using this mechanism, but the list is in no sense complete. This brings us to our final topic, future developments and support.

Development-Support Groups

IBM, through its ACIS organization, has contracted with Western Washington University for a three-year development of tools for the creation of CAI materials. These tools will be built around IBM PILOT. The staff, listed to give an idea of the magnitude of the project, will include a systems programmer, a graphic designer, and a lesson designer-PILOT programmer. A number of student programmers and faculty projects will be supported as well. The items described in the preceding section give some idea of the types of developments which will be carried out. Other activities may include development of text editors which handle user-defined characters, interfaces with graphic production devices, and documentation of programming techniques. A major project will be the design of an authoring front end which will provide all the features of a sophisticated authoring system but which will generate PILOT code which can be further developed to provide features beyond the capability of the front end. No doubt additional possibilities will arise as work continues and as feedback from PILOT users is analyzed. We encourage users in this audience to contribute such feedback, and we welcome

both long and short term visitors to the development center.

Finally, we must emphasize that IBM's support of these developments (which developments are driven in large part by author's requirements) in no way implies that the solutions arrived at will become IBM products or that new versions of IBM PILOT will be forthcoming. On the other hand, the additional capabilities must be available to authors if they are to be worthwhile. For the forseeable future they will be distributed, where appropriate, as irregular updates to subscribers to the Advanced Feature Library.

Acknowledgments

The authors wish to acknowledge the input of the many PILOT users whose suggestions have led to the design of IBM PILOT and items in the Advanced Feature Library. There are far too many to acknowledge individually. We also wish to acknowledge the support of IBM; by supporting the development-support center they are raising small machine author tools to a new level of sophistication and usefulness.